6400/6600

SYSTEMS BULLETIN

3

10 October 1966

CONTENTS

FIGURES AND TABLES

6400/6600 SYSTEMS AVAILABILITY REPORT

The systems listed below are currently available from Program Distribution in Palo Alto. Please submit requests to your local CONTROL DATA representative.

CHIPPEWA OPERATING SYSTEM

System	Version	Maint. Doc. Available	Price Per Copy
Chippewa Operating System* (Specify 3000 or 6000 controllers)	1.1	X	3.76
FORTRAN*	1.1		
ASCENT*	1.1	X	5.20
CDCKWIC	1.1	X	.75
MATRIX ALGEBRA SUBROUTINES	1.0	X	.52
PERT/TIME**	1.0	X	1.50

*The system library and source are distributed on one reel of tape. Please reference 64/6600 Systems Bulletin 2 for a description of the contents of the tape. 64/6600 Systems Bulletin 2 is part of the materials distributed with the above systems.

**Please specify when ordering that this product is to be used with Chippewa Operating System.

SCOPE

System	Version	Maint. Doc. Available	Price Per Copy
SCOPE*	2.0		
ASCENT/ASPER*	2.0		
FORTRAN*	2.0		
COPYN*	1.0		
PERT/TIME**	1.0		

The above systems were tested on a 6400 updated through Engineering Change Order (ECO) number 39.

*The system library and source are distributed on one reel of tape. Please reference 64/6600 Systems Bulletin 3 for a description of the contents of the tape. 64/6600 Systems Bulletin 3 is part of the materials distributed with the above systems.

**Please specify when ordering that this product is to be used with SCOPE.

SCOPE Version 2.0 along with ASCENT Version 2.0, FORTRAN Version 2.0 and COPYN Version 1.0 has now been released. The release consists of:

Verification decks for SCOPE Version 2.0, ASCENT Version 2.0, FORTRAN Version 2.0 and COPYN Version 1.0.

A master tape containing the following nine files recorded in binary mode:

File 1. The system file (binary)

File 2. System routines (COSY) (STL, DSD, MTR, CMR)

File 3. CM resident CP routines (COSY)

File 4. CM resident PP routines (COSY)

File 5. Disk resident PP routines (COSY)

File 6. Disk resident utility routines (COSY)

File 7. Disk resident FORTRAN object time routines (COSY)

File 8. ASCENT (COSY)

File 9. FORTRAN (COSY)

RELEASE DESCRIPTION

The System File

Each routine in the system appears on the system file as one or more binary logical records. These binary records are separated by zero length records into logical groups. RSL (and RPL) is no longer one logical record nor are the zero word terminators (3,7,9 card) used.

Group 1. System Routines (STL, DSD, MTR, CMR)

Group 2. Resident Subroutine Library (RSL)

Group 3. Resident Peripheral Library (RPL)

Group 4. Peripheral Library Directory Routines (PLD)

Group 5. Central Library Directory Routines (CLD)

The ordering of routines within these groups is given in Figure 1.

The system file is represented as a binary card deck as follows:

STL 7-8-9 card DSD 7-8-9 card MTR 7-8-9 card CMR 7-8-9 card 7-8-9 card	system routines
ACGOER 7-8-9 card DBLE 7-8-9 card . . . TAN 7-8-9 card XRCL 7-8-9 card 7-8-9 card	RSL
1AJ 7-8-9 card 1BJ 7-8-9 card . . . MSG 7-8-9 card 7-8-9 card	RPL
007 7-8-9 card 1CO 7-8-9 card . . . TIM 7-8-9 card WBR 7-8-9 card 7-8-9 card	PLD
ASCENT 7-8-9 card ASCENT1 7-8-9 card . . . TANH 7-8-9 card TIME 7-8-9 card 7-8-9 card	CLD

Figure 1. Catalog of the System Tape

RECORD	LENGTH	PACKAGE	CKSUM	LENGTH
1	401	STL	2743	401
2	767	DSD	6747	750
3	775	MTR	152	756
4	5017	CMR	7535	5000
5	0			
6	46	ACGOER	7511	27
7	34	DBLE	4061	15
8	114	EXP	1161	75
9	51	GETBA	2036	32
10	64	IBAIEX	3355	45
11	35	LOCF	3345	16
12	144	SINCOS	1617	125
13	33	SNGL	6007	14
14	101	SQRT	4764	62
15	1076	SYSTEM	6300	1057
16	154	TAN	564	135
17	37	XRCL	1006	20
18	0			
19	125	1AJ	35	106
20	151	1BJ	4036	132
21	226	1LJ	4641	207
22	272	1OT	5645	253
23	140	2BD	6534	121
24	133	2BP	406	114
25	73	2CF	352	54
26	107	2DF	1514	70
27	65	2DT	6700	46
28	234	2LP	4643	215
29	223	2RC	4044	204
30	136	2RD	5004	117
31	175	2TB	5653	156
32	126	2TJ	3373	107
33	250	2TR	1441	231
34	340	2TS	1665	321
35	250	2TW	4663	231
36	144	2WD	6720	125
37	54	7DP	1763	35
38	62	7TP	533	43
39	46	CHK	7741	27
40	113	CIO	7056	74
41	45	MSG	6063	26
42	0			
43	1257	007	3334	1240
44	67	1CO	6523	50
45	145	1DF	746	126
46	41	1DS	6541	22
47	35	1FM	172	16
48	133	1LT	672	114
49	35	1PL	16	16
50	273	1PO	2466	254
51	137	1RF	6636	120
52	171	1RI	1176	152
53	145	1RO	4603	126
54	173	1TD	2526	154
55	153	2BT	10	134
56	72	2EF	7563	53
57	330	2LA	6101	311
58	416	2LB	7116	377
59	254	2LE	710	235
60	242	2PC	1440	223
61	222	2RT	326	203
62	217	2WT	3762	200

63	270	3OT	7121	251
64	117	3SD	4000	100
65	104	4SD	403	65
66	1241	DIS	1567	1222
67	675	DMP	666	656
68	45	HLP	204	26
69	121	LBC	6350	102
70	1305	LDR	3767	1266
71	176	LOC	736	157
72	231	LOD	5474	212
73	100	PBC	664	61
74	173	PBS	3123	154
75	52	RBR	36	33
76	43	RFL	6320	24
77	54	SOS	7213	35
78	35	TIM	3275	16
79	53	WBR	6151	34
80	0			
81	265	ASCENT	1632	244
82	11545	ASCENT1	4157	11524
83	265	RUN	3324	244
84	22140	RUN1	530	22117
85	741	Q8DIAGP	4716	720
86	66	BKSP	6072	47
87	426	CATALOG	5722	407
88	113	COPY	1123	74
89	132	COPYBF	3274	113
90	120	COPYSBF	461	101
91	1233	COPYN	41	1214
92	3304	LOADER	7130	3265
93	77	OVERLOD	5443	60
94	54	REWIND	3400	35
95	610	VERIFY	4220	571
96	132	ALNLOG	324	113
97	207	ASINCOS	3473	170
98	135	ATAN	7532	116
99	156	ATAN2	1431	137
100	111	BACKSP	2032	72
101	160	BUFFEI	7532	141
102	137	BUFFEO	2732	120
103	72	CABS	7675	53
104	103	CBAIEX	3712	64
105	103	CCOS	3720	64
106	74	CEXP	337	55
107	71	CLOG	7013	52
108	103	CSIN	1170	64
109	75	CSQRT	1764	56
110	50	DABS	7711	31
111	255	DATAN	4417	236
112	166	DBADEX	5626	147
113	57	DBAIEX	2450	40
114	204	DEXP	2531	165
115	325	DISPLA	676	306
116	251	DLNLOG	7723	232
117	103	DMOD	3223	64
118	55	DSIGN	1467	36
119	247	DSINCOS	1745	230
120	114	DSQRT	5250	75
121	164	DUMP	6744	145
122	36	DVCHK	2630	17
123	76	ENDFIL	6503	57
124	62	IDINT	374	43
125	65	IFENDF	6074	46
126	166	INPUTB	1621	147
127	227	INPUTC	2256	210
128	136	INPUTS	3634	117

129	33	IOCHEC	2643	14
130	200	IOCHEK	5033	161
131	1271	KODER	761	1252
132	1200	KRAKER	3621	1161
133	56	LENGTH	136	37
134	134	OUTPTB	2326	115
135	213	OUTPTC	644	174
136	140	OUTPTS	4777	121
137	35	OVERFL	6045	16
138	107	OVERLAY	1007	70
139	54	PAUSE	5727	35
140	42	RANF	7647	23
141	66	RBAIEX	3370	47
142	124	RBAREX	5340	105
143	56	REMARK	3730	37
144	106	REWINM	713	67
145	55	SECOND	2677	36
146	200	SEGMENT	7745	161
147	50	SLITE	6070	31
148	55	SLITET	6056	36
149	52	SSWTCH	4133	33
150	37	START	3751	20
151	105	TANH	7011	66
152	55	TIME	136	36
153	0			

END OF FILE

The Cosy Files

The content of each of the cosy files is as follows:

File 2. System Routines

1. STL
2. DSD
3. MTR
4. CMR

File 3. CM Resident CP Routines

1. ACGOER
2. DBLE
3. EXP
4. GETBA
5. IBAIEX
6. LOCF
7. SINCOS
8. SNGL
9. SQRT
10. SYSTEM
11. TAN
12. XRCL

File 4. CM Resident PP Routines

1. 1AJ
2. 1BJ
3. 1LJ
4. 1OT
5. 2BD
6. 2BP
7. 2CF
8. 2DF
9. 2DT
10. 2LP
11. 2RC
12. 2RD
13. 2TB
14. 2TJ
15. 2TR
16. 2TS
17. 2TW
18. 2WD
19. 7DP
20. 7TP
21. CHK
22. CIO
23. MSG

File 5. Disk Resident PP Routines

1. 007
2. 1CO
3. 1DF
4. 1DS
5. 1FM
6. 1LT
7. 1PL
8. 1PO
9. 1RF
10. 1RI
11. 1RO
12. 1TD
13. 2BT
14. 2EF
15. 2LA
16. 2LB

17. 2LE
18. 2PC
19. 2RT
20. 2WT
21. 3OT
22. 3SD
23. 4SD
24. DIS
25. DMP
26. HLP
27. LBC
28. LDR
29. LOC
30. LOD
31. PBC
32. PBS
33. RBR
34. RFL
35. SOS
36. TIM
37. WBR

File 6. Disk Resident Utility Routines

1. BKSP
2. CATALOG
3. COPY
4. COPYBF
5. COPYN
6. COPYSBF
7. LOADER
8. OVERLOD
9. REWIND
10. VERIFY

File 7. Disk Resident FORTRAN Object Time Routines

1. ALNLOG
2. ASINCOS
3. ATAN
4. ATAN2
5. BACKSP
6. BUFFEI
7. BUFFEO
8. CABS
9. CBAIEX
10. CCOS
11. CEXP
12. CLOG
13. CSIN
14. CSQRT
15. DABS
16. DATAN
17. DBADEX
18. DBAIEX
19. DEXP
20. DISPLA
21. DLNLOG
22. DMOD
23. DSIGN
24. DSINCOS
25. DSQRT
26. DUMP
27. DVCHK
28. ENDFIL
29. IDINT
30. IFENDF
31. INPUTB

32. INPUTC
33. INPUTS
34. IOCHEC
35. IOCHEK
36. KODER
37. KRAKER
38. LENGTH
39. OUTPTB
40. OUTPTC
41. OUTPTS
42. OVERFL
43. OVERLAY
44. PAUSE
45. RANF
46. RBAIEX
47. RBAREX
48. REMARK
49. REWINM
50. SECOND
51. SEGMENT
52. SLITE
53. SLITET
54. SSWTCH
55. START
56. TANH
57. TIME

File 8. ASCENT

1. ASCENT (overlay 0,0)
2. ASCENT1 (overlay 1,0)

File 9. FORTRAN

1. RUN (overlay 0,0)
2. RUN1 (overlay 1,0)
3. Q8DIAGP (overlay 1,1)

The List Tape

The composite list tape contains the following routines and is written in packed display code:

File 1 System routines (STL, DSD, MTR, CMR)

File 2 CM resident CP routines

File 3 CM resident PP routines

File 4 Disk resident PP routines

File 5 Disk resident utility routines

File 6 Disk resident FORTRAN object time routines

File 7 ASCENT

File 8 FORTRAN

INSTALLATION INSTRUCTIONS

Modifying CMR

The only modifications which may be necessary to the system file are changes to the Equipment Status Table (EST). Memory size is automatically assigned by MTR at dead start time. The EST modifications may be made in a number of ways:

1. Changing EST (locations 2100-2200) from the console after dead start.

2. Use CMR from COSY file with appropriate modification cards.

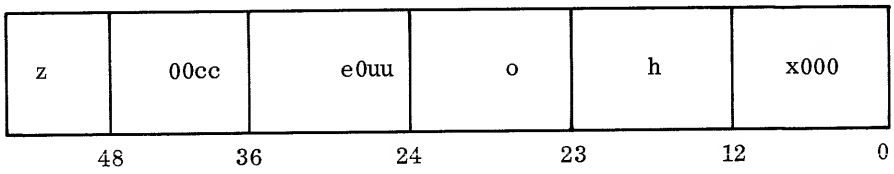
A COSY deck of CMR has been included in this release (file 1 record 4). CMR is composed of sixteen elements in one ASPER program.

		Location (octal)	
1.	POINTERS	0-30	
2.	DATE LINE	31-36	Preset in CMR
3.	START	37-57	
4.	PPCOM	60-177	
5.	CPAREA	200-1777	
6.	CPRES	2000-2077	
7.	EST	2100-2177	
8.	CLD	2200-2377	
9.	PLD	2400-2477	
10.	TRT0	2500-2577	
11.	TRT1	2600-2677	
12.	TRT2	2700-2777	
13.	TRT3	3000-3077	
14.	TRT4	3100-3177	
15.	FNT/FST	aaaa-3777	
16.	DFB	4000-4777	

aaaa may vary from 2600 to 3200 depending on the number of disks in the system. Each disk requires a 100_8 word TRT (TRT0, TRT1, TRT2, TRT3, TRT4). When the system contains fewer than 5 disks, the origin of FNT/FST may be moved back into the space reserved for the unused TRT tables. CMR is provided in ASPER source language. This is necessary in order that the binary text be free of loader tables.

Equipment Status Table (EST)

The format of EST for 6000 equipment is as follows:



z=2000	Signifies an empty EST entry. The remaining bytes are zero.
=0000	Signifies the entry defines a piece of equipment in the system. The remaining bytes are significant.
cc	Channel on which the equipment is attached.
e	6000 synchronizer number.
uu	Unit number
o	On/off bit; 0 indicates off, 1 indicates on. This bit can be changed with the ONnn/OFFnn statements from the console.
h	Equipment type in display code:

DA	Channel 0 disk unit
DB	Channel 1 disk unit
DC	Channel 2 disk unit
DD	Channel 3 disk unit
DE	Channel 4 disk unit
CR	Card reader
CP	Card punch
DS	Display console
LP	Line printer
MT	607 magnetic tape
WT	626 magnetic tape

x	Zero indicates 6000 equipment.

The format of EST for 3000 equipment is as follows:

Z	BB AA	DD CC	O HH	SEUU

where Z = 2000 Signifies an empty EST entry. The remaining bytes are zero. (12 bits)

= 0000 Signifies the entry defines a piece of equipment.

AA, BB, CC, DD are channels connected. (6 bits each)

O is the on/off bit. (1 bit)

HH is the equipment type (11 bits) in display code as listed above.

S is the 6681 number. (3 bits)

E is the equipment number. (3 bits)

UU is the unit number. (6 bits).

```
                                ASPER  CMR                                                    00001
                                ,*************************************************************00002
                                ,                                                             00003
                                ,       CMR -- CENTRAL MEMORY RESIDENT                        00004
                                ,                                                             00005
                                ,*************************************************************00006
                                ,                                                             00007
                                ,       ALL CENTRAL MEMORY TABLES ARE POSITIONED              00008
                                ,        AND POINTERS ARE SET ACCORDING TO                    00009
                                ,        THE STARTING ADDRESSES SET BELOW --                  00010
                                ,                                                             00011
000200                  CPAREA  EQU     200B            ,CONTROL POINT AREAS                  00012
002000                  CPRES   EQU     2000B           ,CP RESIDENTS                         00013
002100                  EST     EQU     2100B           ,EQUIPMENT STATUS TABLE               00014
002200                  CLD     EQU     2200B           ,CENTRAL LIBRARY DIRECTORY            00015
002400                  PLD     EQU     2400B           ,PERIPHERAL LIBRARY DIRECTORY         00016
002500                  TRT0    EQU     2500B           ,TRACK RESERVATION TABLE -- DISK 0    00017
003200                  FNT     EQU     3200B           ,FILE NAME + STATUS TABLE             00018
004000                  DFB     EQU     4000B           ,DAYFILE BUFFER                       00019
005000                  RSL     EQU     5000B           ,RESIDENT CP SUBROUTINE LIBRARY       00020
007000                  RPL     EQU     7000B           ,RESIDENT PERIPHERAL LIBRARY          00021
                                ,                                                             00022
                                ,                                                             00023
004003                  DFBIN   EQU     DFB+3           ,INPUT POINTER FOR DFB                00024
002600                  TRT1    EQU     TRT0+100B       ,TRACK RESERVATION TABLE -- DISK 1    00025
002700                  TRT2    EQU     TRT1+100B       ,TRACK RESERVATION TABLE -- DISK 2    00026
003000                  TRT3    EQU     TRT2+100B       ,TRACK RESERVATION TABLE -- DISK 3    00027
003100                  TRT4    EQU     TRT3+100B       ,TRACK RESERVATION TABLE -- DISK 4    00028
007777                  LTRK    EQU     7777B           ,LAST TRACK NO.  (DISK POSITION)      00029
000100                  SLOZ    EQU     0100B           ,SECTOR LIMIT FOR OUTER ZONE HALF-TRACKS 00030
000062                  SLIZ    EQU     0062B           ,SECTOR LIMIT FOR INNER ZONE HALF-TRACKS 00031
                                ,                                                             00032
                                ,       POINTERS TO CM TABLES                                 00033
                                ,                                                             00034
0000  0315                     CON     0315B,2200B,0,0,5000B  SYSTEM LABEL --CMR,            00035
0001  2200
0002  0000
0003  0000
0004  5000
0005  7000                     CON     RPL,0,0,0,0            RPL POINTER,                   00036
0006  0000
0007  0000
0010  0000
0011  0000
0012  2400                     CON     PLD,TRT0,0,0,0         PLD POINTER,                   00037
0013  2500
0014  0000
0015  0000
0016  0000
0017  4000                     CON     DFB,DFBIN,DFB,RSL,0    DFB POINTER,                   00038
0020  4003
0021  4000
0022  5000
0023  0000
0024  3200                     CON     FNT,DFB,0,0,0          FNT POINTER,                   00039
0025  4000
```


Figure 2. Listing of CMR

```
0026 0000
0027 0000
0030 0000
0031 2100   CON   EST,CLD,0,0,0          EST POINTER.                    00040
0032 2200
0033 0000
0034 0000
0035 0000
0036 5000   CON   RSL,RPL,0,0,0          RSL POINTER.                    00041
0037 7000
0040 0000
0041 0000
0042 0000
0043 2200   CON   CLD,PLD,0,0,0          CLD POINTER.                    00042
0044 2400
0045 0000
0046 0000
0047 0000
0050 2500   CON   TRT0,LTRK,0,SLOZ,SLIZ  TRT DISK 0.                     00043
0051 7777
0052 0000
0053 0100
0054 0062
0055 2600   CON   TRT1,LTRK,0,SLOZ,SLIZ  TRT DISK 1.                     00044
0056 7777
0057 0000
0060 0100
0061 0062
0062 2700   CON   TRT2,LTRK,0,SLOZ,SLIZ  TRT DISK 2.                     00045
0063 7777
0064 0000
0065 0100
0066 0062
0067 3000   CON   TRT3,LTRK,0,SLOZ,SLIZ  TRT DISK 3.                     00046
0070 7777
0071 0000
0072 0100
0073 0062
0074 3100   CON   TRT4,LTRK,0,SLOZ,SLIZ  TRT DISK 4.                     00047
0075 7777
0076 0000
0077 0100
0100 0062
0101 0000   BSSZ  15                     CHANNEL STATUS TABLE(CST).      00048
0120 0003   CON   3,0,0,0,0              STATUS FOR PSEUDO-CONTROL POINT. 00049
0121 0000
0122 0000
0123 0000
0124 0000
0125 1517   CON   1517B,1611B,2417B,2200B,0  JOB NAME FOR CONTROL POINT 0 00050
0126 1611
0127 2417
0130 2200
0131 0000
0132 0000   BSSZ  25                     IDLE TIMES.                     00051
0163 0001   CON   1,0,0,0,0              INITIAL P ADR. FOR SIMULATOR.   00052
```

```
0164  0000
0165  0000
0166  0000
0167  0000
0170  5533      DPC   • 00,00,00,•        TIME                                  00053
0171  3357
0172  3333
0173  5733
0174  3357
0175  5523      DPC   • SCOPE OPERATING SYSTEM • VERSION 2,0, JULY 1966  •      00054
0176  0317
0177  2005
0200  5517
0201  2005
0202  2201
0203  2411
0204  1607
0205  5523
0206  3123
0207  2405
0210  1555
0211  4655
0212  2605
0213  2223
0214  1117
0215  1655
0216  3557
0217  3356
0220  5512
0221  2514
0222  3155
0223  3444
0224  4141
0225  5555
0226  0000      CON   0,0,0,0,0                                                 00055
0227  0000
0230  0000
0231  0000
0232  0000
0233  0000      BSSZ  490                 STARTING TIMES,COMUNICATION AREA      00056
1205  0001      CON   1,4000B,0,0,0       RA=CONTROL POINT  ,                   00057
1206  4000
1207  0000
1210  0000
1211  0000
1212  0000      BSSZ  70                  EXCHANGE JUMP PACKAGE                 00058
1320  0000      CON   0,0,0,140B,0                                              00059
1321  0000
1322  0000
1323  0140
1324  0000
1325  0000      BSSZ  500                                                       00060
2405  0001      CON   1,4000B,0,0,0       RA=CONTROL POINT  ,                   00061
2406  4000
2407  0000
2410  0000
```

```
2411  0000
2412  0000
2520  0000
2521  0000
2522  0000
2523  0140
2524  0000
2525  0000
3605  0001
3606  4000
3607  0000
3610  0000
3611  0000
3612  0000
3720  0000
3721  0000
3722  0000
3723  0140
3724  0000
3725  0000
5005  0001
5006  4000
5007  0000
5010  0000
5011  0000
5012  0000
5120  0000
5121  0000
5122  0000
5123  0140
5124  0000
5125  0000
6205  0001
6206  4000
6207  0000
6210  0000
6211  0000
6212  0000
6320  0000
6321  0000
6322  0000
6323  0140
6324  0000
6325  0000
7405  0001
7406  4000
7407  0000
7410  0000
7411  0000
7412  0000
7520  0000
7521  0000
7522  0000
7523  0140
7524  0000
7525  0000
```

```
      BSSZ  70                                                    00062
      CON   0,0,0,140B,0        EXCHANGE JUMP PACKAGE             00063

      BSSZ  560                                                   00064
      CON   1,4000B,0,0,0       RA=CONTROL POINT  ,               00065

      BSSZ  70                                                    00066
      CON   0,0,0,140B,0        EXCHANGE JUMP PACKAGE             00067

      BSSZ  560                                                   00068
      CON   1,4000B,0,0,0       RA=CONTROL POINT  ,               00069

      BSSZ  70                                                    00070
      CON   0,0,0,140B,0        EXCHANGE JUMP PACKAGE             00071

      BSSZ  560                                                   00072
      CON   1,4000B,0,0,0       RA=CONTROL POINT  ,               00073

      BSSZ  70                                                    00074
      CON   0,0,0,140B,0        EXCHANGE JUMP PACKAGE             00075

      BSSZ  560                                                   00076
      CON   1,4000B,0,0,0       RA=CONTROL POINT  ,               00077

      BSSZ  70                                                    00078
      CON   0,0,0,140B,0        EXCHANGE JUMP PACKAGE             00079

      BSSZ  560                                                   00080
```

```
Z        0605    0001          CON     1,4000B,0,0,0          RA=CONTROL POINT  .                00081
Z        0606    4000
Z        0607    0000
Z        0610    0000
Z        0611    0000
Z        0612    0000          BSSZ    70                     EXCHANGE JUMP PACKAGE              00082
Z        0720    0000          CON     0,0,0,140B,0                                              00083
Z        0721    0000
Z        0722    0000
Z        0723    0140
Z        0724    0000
Z        0725    0000          BSSZ    555                                                       00084
Z      002022                  MOVE    EQU     2022B                                             00085
Z        2000    0000          CON     0,MOVE,0,0,0           START OF STORAGE MOVE PROGRAM,     00086
Z        2001    2022
Z        2002    0000
Z        2003    0000
Z        2004    0000          BSSZ    5                                                         00087
Z        2005    0000          CON     40B,0,0,0,0            SET EXIT MODE,                     00088
Z        2012    0040
Z        2013    0000
Z        2014    0000
Z        2015    0000
Z        2016    0000          BSSZ    75                     EXCHANGE JUMP PACKAGE,             00089
Z        2017    0000  C       * STORAGE MOVE PROGRAM *                                          00090
Z                      CENTRY  EQ      B1=B2,EXIT   $  SB7 =1                                    00091
Z                       ENTRY  CON     0412B,0,61B,7000B,1                                       00092
Z        2132    0412
Z        2133    0000
Z        2134    0061
Z        2135    7000
Z        2136    0001  C       NG      B5,DOWN  $ SB5 =-2                                       00093
Z                      CON     730B,20B,2561B,5077B,7775B                                        00094
Z        2137    0730
Z        2140    0020
Z        2141    2561
Z        2142    5077
Z        2143    7775  C       SA1  B2-B7 $ SA2 B2+B5 $ JP LOOP                                  00095
Z                      CON     5712B,7562B,2502B,0,2026B                                         00096
Z        2144    5712
Z        2145    7562
Z        2146    2502
Z        2147    0000
Z        2150    2026          CDOWN   SB5 B7+B7 $ SA1 B1 $ SA2 B1+B7                            00097
Z                      CON     6657B,7561B,1056B,2174B,6000B                                     00098
Z        2151    6657
Z        2152    7561
Z        2153    1056
Z        2154    2174
Z        2155    6000          CLOOP   SA3 A1+B5 $ SA4 A2+B5 $ BX6 X1 $ LX7 X2                  00099
Z                      CON     5431B,5544B,2510B,6102B,2702B                                     00100
Z        2156    5431
Z        2157    5544
Z        2160    2510
Z        2161    6102
Z        2162    2702
```

```
Z                                C   SA6 A1+B6 $ SA7 A2+B3 $ BX6 X3 $ LX7 X4                00101
Z       2163    5461                 CON     5+61B,3547B,2310B,6302B,2704B                  00102
Z       2164    3547
Z       2165    2310
Z       2166    6302
Z       2167    2704
Z                                C   SA6 A3+B6 $ SA7 A4+B3 $ SB1 B1+4                      00103
Z       2170    5463                 CON     5463B,3547B,4361B,1100B,4                      00104
Z       2171    3547
Z       2172    4361
Z       2173    1100
Z       2174    0004
Z                                C   SA1 A3+B6 $ SA2 A4+B5 $ LT B1,B2 LOOP                 00105
Z       2175    5413                 CON     5413B,5542B,4507B,1200B,2026B                  00106
Z       2176    5542
Z       2177    4507
Z       2200    1200
Z       2201    2026
Z                                C   JP EXIT                                               00107
Z       2202    0200                 CON     0200B,0,0,0,0                                  00108
Z       2203    0000
Z       2204    0000
Z       2205    0000
Z       2206    0000
Z       2207    0000                 BSSZ    25                                             00109
Z       2240    0000                 CON     0,2,0,0,0                                      00110
Z       2241    0002
Z       2242    0000
Z       2243    0000
Z       2244    0000
Z       2245    0000                 CON     0,2060B,0,0,0                                  00111
Z       2246    2060
Z       2247    0000
Z       2250    0000
Z       2251    0000
Z       2252    0000                 CON     0,20B,0,0,0                                    00112
Z       2253    0020
Z       2254    0000
Z       2255    0000
Z       2256    0000
Z       2257    0000                 BSSZ    65                                             00113
Z       2360    3333                 CON     3333B,3300B,0,0,0                              00114
Z       2361    3300
Z       2362    0000
Z       2363    0000
Z       2364    0000
Z       2365    0000                 CON     0,0,0,0,0                                      00115
Z       2366    0000
Z       2367    0000
Z       2370    0000
Z       2371    0000
Z       2372    0400                 CON     0400B,0,0200B,0,0                              00116
Z       2373    0000
Z       2374    0200
Z       2375    0000
Z       2376    0000
```

Z 2377 0000 BSSZ 65 00117

```
Z                                  .*************************************************************
Z                                  .
Z                                  .       EST -- EQUIPMENT STATUS TABLE
Z                                  .
Z                                  .*************************************************************
Z                                  .
Z                                  .       FORMAT FOR 3000-SERIES CONTROLLERS
Z                                  .
Z                                  .  1ST BYTE = CONTROL PT. ADDRESS (TO WHICH EQUIP IS ASSIGNED)
Z                                  .  2ND BYTE = A + B CHANNELS CONNECTED INTO CONTROLLER
Z                                  .  3RD BYTE = C + D CHANNELS CONNECTED INTO CONTROLLER
Z                                  .  4TH BYTE = EQUIP. TYPE (2 LETTERS IN DISPLAY CODE)
Z                                  .  4TH BYTE, LEFTMOST BIT = INTERLOCK (0 = ON, 1 = OFF)
Z                                  .  5TH BYTE = 6681 NO. + 3000 EQUIP. NO. + UNIT NO.
Z                                  .
Z                                  .*************************************************************
Z                                  .
Z                                  .       FORMAT FOR 6000 SYNCHRONIZERS
Z                                  .
Z                                  .  1ST BYTE = CONTROL POINT ADDRESS,  OR 2000 IF EMPTY ENTRY
Z                                  .  2ND BYTE = CHANNEL NO.
Z                                  .  3RD BYTE = SYNCHRONIZER + UNIT NO.
Z                                  .  4TH BYTE = INTERLOCK + EQUIP. TYPE IN DISPLAY CODE
Z                                  .  5TH BYTE = NOT USED
Z                                  .
Z                                  .*************************************************************
Z    000000               IN       EQU    0            .EQUIPMENT IS IN THIS CONFIGURATION
Z    002000               OUT      EQU    2000B        .EMPTY EST ENTRY -- NO EQUIP. ATTACHED
Z    000000               DACH     EQU    0            .DISK 0 (DA) CHANNEL = 0
Z    001000               SYNC     EQU    1000B        .DISK SYNCHRONIZER = 1, UNIT NO. = 0
Z    000012               CR1C     EQU    12B          .CARD READER 1 ON CHAN 12B
Z    000400               CR1E     EQU    400B         .CARD READER 1 = EQUIP. 4
Z    000013               CPCH     EQU    13B          .CARD PUNCH ON CHAN. 13
Z    000700               CPEQ     EQU    700B         .CARD PUNCH = EQUIP. 7
Z    000010               DS1C     EQU    10B          .DISPLAY SCOPE 1 ON CHAN. 10
Z    007000               DSYN     EQU    7000B        .DISPLAY SCOPE SYNCHRONIZER = 7
Z    000011               LP1C     EQU    11B          .LINE PRINTER 1 ON CHAN. 11
Z    000600               LP1E     EQU    600B         .LINE PRINTER 1 = EQUIP. 6
Z    000011               LP2C     EQU    11B          .LINE PRINTER 2 ON CHAN. 11
Z    000700               LP2E     EQU    700B         .LINE PRINTER 2 = EQUIP. 7
Z    000012               MTCH     EQU    12B          .MAG. TAPE CONTROLLER ON CHANNEL 12
Z    000500               MT1E     EQU    500B         .MT1 = EQUIP. 5, UNIT 0
Z    000501               MT2E     EQU    501B         .MT2 = EQUIP. 5, UNIT 1
Z    000502               MT3E     EQU    502B         .MT3 = EQUIP. 5, UNIT 2
Z    000503               MT4E     EQU    503B         .MT4 = EQUIP. 5, UNIT 3
Z                                  .
Z    000401               DA       EQU    0401B
Z    000322               CR       EQU    0322B
Z    000320               CP       EQU    0320B
Z    000423               DS       EQU    0423B
Z    001420               LP       EQU    1420B
Z    001524               MT       EQU    1524B
Z      2500     0000               CON    IN,DACH,SYNC,DA,0      DISK 0 ON CHAN 0.
Z      2501     0000
Z      2502     1000
Z      2503     0401
```

```
Z        2504    0000
Z        2505    2000    CON     OUT,0,0,0,0                                     00172
Z        2506    0000
Z        2507    0000
Z        2510    0000
Z        2511    0000
Z        2512    2000    CON     OUT,0,0,0,0                                     00173
Z        2513    0000
Z        2514    0000
Z        2515    0000
Z        2516    0000
Z        2517    2000    CON     OUT,0,0,0,0                                     00174
Z        2520    0000
Z        2521    0000
Z        2522    0000
Z        2523    0000
Z        2524    2000    CON     OUT,0,0,0,0                                     00175
Z        2525    0000
Z        2526    0000
Z        2527    0000
Z        2530    0000
Z        2531    0000    CON     IN,CR1C,0,CR,CR1E       CARD READER1,           00176
Z        2532    0012
Z        2533    0000
Z        2534    0322
Z        2535    0400
Z        2536    2000    CON     OUT,0,0,0,0                                     00177
Z        2537    0000
Z        2540    0000
Z        2541    0000
Z        2542    0000
Z        2543    0000    CON     IN,CPCH,0,CP,CPEQ       CARD PUNCH,             00178
Z        2544    0013
Z        2545    0000
Z        2546    0320
Z        2547    0700
Z        2550    0000    CON     IN,DS1C,DSYN,DS,0       DISPLAY SCOPE,          00179
Z        2551    0010
Z        2552    7000
Z        2553    0423
Z        2554    0000
Z        2555    2000    CON     OUT,0,0,0,0                                     00180
Z        2556    0000
Z        2557    0000
Z        2560    0000
Z        2561    0000
Z        2562    2000    CON     OUT,0,0,0,0                                     00181
Z        2563    0000
Z        2564    0000
Z        2565    0000
Z        2566    0000
Z        2567    2000    CON     OUT,0,0,0,0                                     00182
Z        2570    0000
Z        2571    0000
Z        2572    0000
Z        2573    0000
```

```
Z    2574  2000    CON   OUT,0,0,0,0                                    00183
Z    2575  0000
Z    2576  0000
Z    2577  0000
Z    2600  0000
Z    2601  2000    CON   OUT,0,0,0,0                                    00184
Z    2602  0000
Z    2603  0000
Z    2604  0000
Z    2605  0000
Z    2606  2000    CON   OUT,0,0,0,0                                    00185
Z    2607  0000
Z    2610  0000
Z    2611  0000
Z    2612  0000
Z    2613  2000    CON   OUT,0,0,0,0                                    00186
Z    2614  0000
Z    2615  0000
Z    2616  0000
Z    2617  0000
Z    2620  0000    CON   IN,LP1C,0,LP,LP1E      LINE PRINTER 1,       00187
Z    2621  0011
Z    2622  0000
Z    2623  1420
Z    2624  0600
Z    2625  0000    CON   IN,LP2C,0,LP,LP2E      LINE PRINTER 2,       00188
Z    2626  0011
Z    2627  0000
Z    2630  1420
Z    2631  0700
Z    2632  2000    CON   OUT,0,0,0,0                                    00189
Z    2633  0000
Z    2634  0000
Z    2635  0000
Z    2636  0000
Z    2637  2000    CON   OUT,0,0,0,0                                    00190
Z    2640  0000
Z    2641  0000
Z    2642  0000
Z    2643  0000
Z    2644  2000    CON   OUT,0,0,0,0                                    00191
Z    2645  0000
Z    2646  0000
Z    2647  0000
Z    2650  0000
Z    2651  2000    CON   OUT,0,0,0,0                                    00192
Z    2652  0000
Z    2653  0000
Z    2654  0000
Z    2655  0000
Z    2656  2000    CON   OUT,0,0,0,0                                    00193
Z    2657  0000
Z    2660  0000
Z    2661  0000
Z    2662  0000
Z    2663  2000    CON   OUT,0,0,0,0                                    00194
```

```
Z        2664    0000
Z        2665    0000
Z        2666    0000
Z        2667    0000
Z        2670    2000        CON     OUT,0,0,0,0                     00195
Z        2671    0000
Z        2672    0000
Z        2673    0000
Z        2674    0000
Z        2675    2000        CON     OUT,0,0,0,0                     00196
Z        2676    0000
Z        2677    0000
Z        2700    0000
Z        2701    0000
Z        2702    2000        CON     OUT,0,0,0,0                     00197
Z        2703    0000
Z        2704    0000
Z        2705    0000
Z        2706    0000
Z        2707    2000        CON     OUT,0,0,0,0                     00198
Z        2710    0000
Z        2711    0000
Z        2712    0000
Z        2713    0000
Z        2714    2000        CON     OUT,0,0,0,0                     00199
Z        2715    0000
Z        2716    0000
Z        2717    0000
Z        2720    0000
Z        2721    2000        CON     OUT,0,0,0,0                     00200
Z        2722    0000
Z        2723    0000
Z        2724    0000
Z        2725    0000
Z        2726    2000        CON     OUT,0,0,0,0                     00201
Z        2727    0000
Z        2730    0000
Z        2731    0000
Z        2732    0000
Z        2733    2000        CON     OUT,0,0,0,0                     00202
Z        2734    0000
Z        2735    0000
Z        2736    0000
Z        2737    0000
Z        2740    2000        CON     OUT,0,0,0,0                     00203
Z        2741    0000
Z        2742    0000
Z        2743    0000
Z        2744    0000
Z        2745    2000        CON     OUT,0,0,0,0                     00204
Z        2746    0000
Z        2747    0000
Z        2750    0000
Z        2751    0000
Z        2752    2000        CON     OUT,0,0,0,0                     00205
Z        2753    0000
```

```
Z    2754  0000
Z    2755  0000
Z    2756  0000
Z    2757  2000   CON   OUT-0,0,0,0                                  00206
Z    2760  0000
Z    2761  0000
Z    2762  0000
Z    2763  0000
Z    2764  2000   CON   OUT,0,0,0,0                                  00207
Z    2765  0000
Z    2766  0000
Z    2767  0000
Z    2770  0000
Z    2771  2000   CON   OUT,0,0,0,0                                  00208
Z    2772  0000
Z    2773  0000
Z    2774  0000
Z    2775  0000
Z    2776  2000   CON   OUT,0,0,0,0                                  00209
Z    2777  0000
Z    3000  0000
Z    3001  0000
Z    3002  0000
Z    3003  2000   CON   OUT,0,0,0,0                                  00210
Z    3004  0000
Z    3005  0000
Z    3006  0000
Z    3007  0000
Z    3010  0000   CON   IN,MTCH,0,MT,MT1E      MAG TAPE 1,          00211
Z    3011  0012
Z    3012  0000
Z    3013  1524
Z    3014  0500
Z    3015  0000   CON   IN,MTCH,0,MT,MT2E      MAG TAPE 2,          00212
Z    3016  0012
Z    3017  0000
Z    3020  1524
Z    3021  0501
Z    3022  0000   CON   IN,MTCH,0,MT,MT3E      MAG TAPE 3,          00213
Z    3023  0012
Z    3024  0000
Z    3025  1524
Z    3026  0502
Z    3027  0000   CON   IN,MTCH,0,MT,MT4E      MAG TAPE 4,          00214
Z    3030  0012
Z    3031  0000
Z    3032  1524
Z    3033  0503
Z    3034  2000   CON   OUT,0,0,0,0                                 00215
Z    3035  0000
Z    3036  0000
Z    3037  0000
Z    3040  0000
Z    3041  2000   CON   OUT,0,0,0,0                                 00216
Z    3042  0000
Z    3043  0000
```

```
Z       3044    0000
Z       3045    0000
Z       3046    2000        CON     OUT,0,0,0,0                     00217
Z       3047    0000
Z       3050    0000
Z       3051    0000
Z       3052    0000
Z       3053    2000        CON     OUT,0,0,0,0                     00218
Z       3054    0000
Z       3055    0000
Z       3056    0000
Z       3057    0000
Z       3060    2000        CON     OUT,0,0,0,0                     00219
Z       3061    0000
Z       3062    0000
Z       3063    0000
Z       3064    0000
Z       3065    2000        CON     OUT,0,0,0,0                     00220
Z       3066    0000
Z       3067    0000
Z       3070    0000
Z       3071    0000
Z       3072    2000        CON     OUT,0,0,0,0                     00221
Z       3073    0000
Z       3074    0000
Z       3075    0000
Z       3076    0000
Z       3077    2000        CON     OUT,0,0,0,0                     00222
Z       3100    0000
Z       3101    0000
Z       3102    0000
Z       3103    0000
Z       3104    2000        CON     OUT,0,0,0,0                     00223
Z       3105    0000
Z       3106    0000
Z       3107    0000
Z       3110    0000
Z       3111    2000        CON     OUT,0,0,0,0                     00224
Z       3112    0000
Z       3113    0000
Z       3114    0000
Z       3115    0000
Z       3116    2000        CON     OUT,0,0,0,0                     00225
Z       3117    0000
Z       3120    0000
Z       3121    0000
Z       3122    0000
Z       3123    2000        CON     OUT,0,0,0,0                     00226
Z       3124    0000
Z       3125    0000
Z       3126    0000
Z       3127    0000
Z       3130    2000        CON     OUT,0,0,0,0                     00227
Z       3131    0000
Z       3132    0000
Z       3133    0000
```

```
Z     3134  0000
Z     3135  2000      CON     OUT,0,0,0,0                                    00228
Z     3136  0000
Z     3137  0000
Z     3140  0000
Z     3141  0000
Z     3142  2000      CON     OUT,0,0,0,0                                    00229
Z     3143  0000
Z     3144  0000
Z     3145  0000
Z     3146  0000
Z     3147  2000      CON     OUT,0,0,0,0                                    00230
Z     3150  0000
Z     3151  0000
Z     3152  0000
Z     3153  0000
Z     3154  2000      CON     OUT,0,0,0,0                                    00231
Z     3155  0000
Z     3156  0000
Z     3157  0000
Z     3160  0000
Z     3161  2000      CON     OUT,0,0,0,0                                    00232
Z     3162  0000
Z     3163  0000
Z     3164  0000
Z     3165  0000
Z     3166  2000      CON     OUT,0,0,0,0                                    00233
Z     3167  0000
Z     3170  0000
Z     3171  0000
Z     3172  0000
Z     3173  2000      CON     OUT,0,0,0,0                                    00234
Z     3174  0000
Z     3175  0000
Z     3176  0000
Z     3177  0000
Z     3200  0000      BSSZ    2960                                           00235
Z     0200  0401      CON     0401B,3106B,1114B,0500B,0020B    DAYFILE       00236
Z     0201  3106
Z     0202  1114
Z     0203  0500
Z     0204  0020
Z     0205  0000      BSSZ    1915                                           00237
Z     4000  5533      UPC     * 00.00.00, DEAD-STAR*  INITIAL DAYFILE ENTRY  00238
Z     4001  3357
Z     4002  3333
Z     4003  5733
Z     4004  3357
Z     4005  5504
Z     4006  0501
Z     4007  0446
Z     4010  2324
Z     4011  0122
Z     4012  2455      CON     2455B,0,0,0,0                                  00239
Z     4013  0000
Z     4014  0000
```

```
   BSSZ   2545                                                    00240
   END                                                            00241
```

508

```
00,00,59. ACMR000. READ.
00,00,59. ACMR000. PP 000 SEC.
00,00,59. ACMR000. ACMR,10,1000,70000.
00,01,13. ACMR000. REQUEST COSTAPE.
00,01,13. ACMR000.  (52 ASSIGNED)
00,01,13. ACMR000. COPYN(,DISC,COSTAPE)
00,01,23. ACMR000. ASCENT(LIST,CI,DISC)
00,01,27. ACMR000.       508  ERRORS IN CMR
00,01,28. ACMR000. CP  008,238 SEC.
00,01,28. ACMR000. PP  013,108 SEC.
SCOPE OPERATING SYSTEM - VERSION 2.0, JULY 1966
```

Dead Start Panel Settings

The dead start panel is set as follows:

	3000 Tape Controller Version
0001	75xx
0002	77xx
0003	e00u
0004	77xx
0005	0010
0006	77xx
0007	1400
0010	74xx
0011	2001
0012	0000
0013	71xx
0014	0015

where e=controller number, u=unit number, and xx=channel number on which the system tape is mounted. (For 3000 systems, xx may only be channels 12 or 13.)

Word 14 has been changed from that in Version 1.1 to make the panel the same as for the engineer's tape.

Preparing The System File From COSY Files

ASCENT Version 2.0 produces a new COSY format. It accepts as input either the old format or the new. The new format allows for COSY output on files other than P80C. The COSY decks in this release are in the new format and must be assembled with Version 2.0. The ASCENT control card and identification of COSY alter cards have also been changed to allow for more flexible file manipulation.

All binary and COSY decks produced by ASCENT and RUN contain an identification header before the binary text. The purpose of this header is to provide a uniform means for identifying the program. The ID header has the following format:

word 1-- 7700 0016 0000 0000 0000B

word 2-- seven or less character name in display code, left justified.

word 3 through 15-- reserved for future use.

COPYN, a new routine provided to aid in library preparation, uses the name in word 2 to identify the logical record. The ID header is stripped off at dead start time by STL. However, STL itself must not contain the header. It must either be stripped off by COPYN or it must be removed physically from a card deck (first card). For decks produced by ASCENT the name is the name found in the operand field of an ASCENT or ASPER pseudo operation. For a FORTRAN program the name is the program name.

FORTRAN and ASCENT are on the library in overlay format. Therefore, whenever changes are made to the programs a new absolute overlay must be generated. This requires assembling the overlay to be corrected, inserting the appropriate overlay card in front of the binary output and then having the loader generate the absolute overlays.

It is possible to create a complete binary system tape without producing binary decks. The following listing shows the control cards along with the associated ASCENT and COPYN directives that are necessary to create such a tape.

```
LBUILD,10,4000,55000.
REQUEST COSY.                          LIBRARY TAPE
REQUEST LIBRARY.                       NEW LIBRARY TAPE
REWIND (COSY)
COPYBF (COSY,XX)                       SKIP OVER BINARY LIBRARY
ASCENT (PO,F1,CI,COSY)                 ASSEMBLE 8 COSY FILES
ASCENT (LGO,F2,CI,COSY)
ASCENT (PO,F3,CI,COSY)
ASCENT (PO,F4,CI,COSY)
ASCENT (LGO,F5,CI,COSY)
ASCENT (LGO,F6,CI,COSY)
ASCENT (LGO,F7,CI,COSY)
ASCENT (LGO,F8,CI,COSY)
COPYN (,F9,F7)                         INSERT OVERLAY DIRECTIVES
COPYN (,F10,F8)
LOAD (F9)                              GENERATE ABSOLUTE OVERLAYS
NOGO.
LOAD(F10)
NOGO.
COPYN (1,LIBRARY,F1)                   STRIP ID FROM STL
COPYN (,LIBRARY,F1)                    GENERATE NEW LIBRARY TAPE
CATALOG (LIBRARY)                      CATALOG NEW TAPE
UNLOAD (LIBRARY)
7-8-9 card                             END OF CONTROL CARDS
        IDENT                          ASCENT DIRECTIVES TO
7-8-9 card                                 ASSEMBLE 8 FILES
        IDENT
7-8-9 card                             COSY MODS TO ROUTINES MAY
        IDENT                              BE INCLUDED HERE.
7-8-9 card
        IDENT
7-8-9 card
        IDENT
7-8-9 card
        IDENT
```

```
7-8-9 card
        IDENT
7-8-9 card
        IDENT
7-8-9 card
REWIND (F7)                          COPYN DIRECTIVES TO INSERT
1,,INPUT                                 OVERLAY CARDS
1,,F7
1,,INPUT
1,,F7
WEOF (F9)
7-8-9 card                           END COPYN DIRECTIVES
OVERLAY(F11,0,0)                     INPUT TO COPYN
7-8-9 card                               ASCENT OVERLAY CARDS
OVERLAY(F11,1,0)
7-8-9 card
REWIND(F8)                           COPYN DIRECTIVES TO INSERT
1,,INPUT                                 OVERLAY CARDS
1,,F8
1,,INPUT
1,,F8
1,,INPUT
1,,F8
WEOF (F10)
7-8-9 card                           END COPYN DIRECTIVES
OVERLAY(F12,0,0)                     INPUT TO COPYN
7-8-9 card                               FORTRAN OVERLAY CARDS
OVERLAY(F12,1,0)
7-8-9 card
OVERLAY(F12,1,1)
7-8-9 card
REWIND (LIBRARY)                     COPYN DIRECTIVES
REWIND (F1)
REWIND (F2)
REWIND (F3)
REWIND (F4)
REWIND (F5)
REWIND (F6)
REWIND (F11)
REWIND (F12)
STL,,F1
7-8-9 card                           END COPYN DIRECTIVES
1,*,F1                               COPY FILE THRU END OF FILE
SKIPF(LIBRARY,-1)                    SKIP BACK OVER FILE MARK
1,,INPUT                             COPY ZERO LENGTH RECORD
1,*,F2
SKIPF(LIBRARY,-1)
1,,INPUT
1,*,F3
SKIPF(LIBRARY,-1)
```

```
1,,INPUT
1,*,F4
SKIPF(LIBRARY,-1)
1,,INPUT
1,*,F11
SKIPF(LIBRARY,-1)
1,*,F12
SKIPF(LIBRARY,-1)
1,*,F5
SKIPF(LIBRARY,-1)
1,*,F6
SKIPF(LIBRARY,-1)
1,,INPUT
WEOF(LIBRARY)
REWIND(LIBRARY)
7-8-9 card
7-8-9 card
7-8-9 card
7-8-9 card
7-8-9 card
7-8-9 card
6-7-8-9 card
```

ıe following example illustrates how to assemble a program (1BJ), with modification, from the master ɔe and generate a modified library.

```
JOB,10,400,60000.
REQUEST MASTER.
COPYN (,NEWCOSY,MASTER)                 FETCH 1BJ
ASCENT (L,PO,XX,CI,NEWCOSY)             ASSEMBLE 1BJ
REQUEST OLDLIB.
REWIND (OLDLIB)
REQUEST NEWLIB.
REWIND (NEWLIB)
COPYN (,NEWLIB, OLDLIB,XX)              MERGE OLDLIB AND 1BJ ONTO NEWLIB
CATALOG (NEWLIB)
UNLOAD (NEWLIB)
7-8-9 card
REWIND (MASTER)                         COPYN DIRECTIVES
SKIPF (MASTER,3)                        1BJ IN 4th FILE
1BJ,,MASTER                             1BJ COSY TO NEWCOSY
REWIND (NEWCOSY)
7-8-9 card                              END COPYN DIRECTIVES
COSY mods                               MODIFICATIONS TO 1JB
    COSY                                COSY (column 11) TERM. MODS
7-8-9 card
1,1AJ, OLDLIB                           COPY ROUTINES UP TO 1BJ
1BJ,,XX                                 COPY NEW 1BJ
2,*,OLDLIB                              SKIP OLD 1BJ,COPY REST
```

```
REWIND (OLDLIB)
REWIND (NEWLIB)
7-8-9 card
6-7-8-9 card
```

This example illustrates how to modify ASCENT1, generate new overlays for ASCENT, and prepare a new library.

```
JOB, 10,200,60000.
REQUEST NEWLIB.
REWIND (NEWLIB)
REQUEST OLDLIB.
REWIND (OLDLIB)
REQUEST MASTER.
REWIND (MASTER)
COPYBF (MASTER,YY,7)                    SKIP 7 FILES.
ASCENT(L,PO,XX,CI,MASTER)               ASSEMBLE ASCENT, ASCENT1
COPYN (,L1,XX)                          INSERT OVERLAY CARDS
LOAD (L1)                               GENERATE OVERLAYS
NOGO.
COPYN (,NEWLIB,OLDLIB,L2)               MERGE OLDLIB AND ASCENT
CATALOG (NEWLIB)
UNLOAD (NEWLIB)
7-8-9 card
    IDENT ASCENT1                       Mods to ASCENT1
    (COSY Mods)
    COSY
    FINIS
7-8-9 card
REWIND (XX)                             COPYN DIRECTIVES
1,,INPUT                                OVERLAYS TO L1
1,,XX
1,,INPUT
1,,XX
7-8-9 card                              END COPYN DIRECTIVES
OVERLAY (L2,0,0)                        INPUT TO COPYN.
7-8-9 card
OVERLAY (L2,1,0)
7-8-9 card
1,ASCENT,OLDLIB                         COPYN DIRECTIVES
SKIPR (NEWLIB,-1)                       COPY UP TO ASCENT
REWIND (L2)
1,2,L2                                  COPY ASCENT, ASCENT1
2,*,OLDLIB                              COPY REST OF FILE
REWIND (OLDLIB)
REWIND (NEWLIB)
7-8-9 card
6-7-8-9 card
```

```
                                        ASCENT - VERSION 2.0                    PAGE NO.     1

                                 ASCENT   DECK1
                                 ENTRY    START
000000  0000000000000000000            CON      0
000001  0000000000000000000            CON      0
000002  0000000000000000000            CON      0
000003  0000000000000000000            CON      0
000004  0000000000000000000    EXIT    CON      0
000005  7160000026           R START   SX6      GENE                SET UP CIO
                5160000022   R         SA6      FIRST
000006  5160000024           R         SA6      OUT
                7170000031   R         SX7      BUFFEND
000007  5170000023           R         SA7      IN
                7170000032   R         SX7      ENDLIM
000010  5170000025           R         SA7      LIMIT
                7160031117             SX6      031117B
000011  20652                          LX6      42
             7150000021      R         SX5      CPB
                      36665            IX6      X6*X5
000012  5160000001                     SA6      1
000013  5110000021           R LOOP    SA1      CPB            TEST FOR END OF CIO
                5120000020   R         SA2      MASK
000014  11312                          BX3      X1*X2
             0303000013      R         ZR       X3,LOOP
000015  5150000017           R         SA5      GETOUT              SET UP THE EXIT
                10650                  BX6      X5
000016  5160000001                     SA6      1
                0400000004   R         EQ       EXIT
000017  0516040000000000000    GETOUT  VFD      D18/END,N42/0
000020  0000000000000000001    MASK    CON      0000000000000000001
000021  1725242025240000024    CPB     CON      172524202524000000248
000022  0000000000000000000    FIRST   CON      0
000023  0000000000000000000    IN      CON      0
000024  0000000000000000000    OUT     CON      0
000025  0000000000000000000    LIMIT   CON      0
000026  5523031720055271722    GENE    DPC      * SCOPE WORKS OK,*
000027  1323551713575555555
000030  0000000000000000000            CON      0
000031  0000000000000000000    BUFFEND CON      0
000032  0000000000000000000    ENDLIM  CON      0
                                 END      START

NUMBER OF LINES WITH DIAGNOSTICS  ---          0
```

Figure 3. SCOPE Verification Program

```
CORE MAP  00.01.48.  NORMAL                    CONTROL                                                    000100    000133    000000    000000
          *--TIME---LOAD MODE --L1--L2-----TYPE-------------USER-------CALL------------FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
          FWA LOADER  125123  FWA TABLES  125111
          *PROGRAM---*ADDRESS*                    --LABELED---COMMON--
          DECK1       000100
          --ENTRY------ADDRESS*                                        REFERENCES
          START       000105
          *----UNSATISFIED EXTERNALS-----                               REFERENCES

SCOPE WORKS OK.
```

```
00.01.42. JOB1001. READ.
00.01.44. JOB1001. PP 002 SEC.
00.01.44. JOB1001. JOB1,17,100,130000.
00.01.45. JOB1001. ASCENT(LIST,PB,LGO,FILE1)
00.01.47. JOB1001. FILE1.
00.01.48. JOB1001. CP  000.440 SEC.
00.01.48. JOB1001. PP  002.422 SEC.
SCOPE OPERATING SYSTEM - VERSION 2.0, JULY 1966
```

```
                                ASCENT
                                ENTRY     TEST

                                MACRO     REVEAL,CC,NN,FF,II,OO,LL
                                SX6       FF
                                SA6       FIRST
                                SX6       II
                                SA6       IN
                                SX6       OO
                                SA6       OUT
                                SX6       LL
                                SA6       LIMIT
                                SA5       CC
                                SX6       NN
                                BX6       X5*X6
                                SA6       CBP
                                SX5       031117B
                                LX5       42
                                SX6       CBP
                                BX6       X5*X6
                                SA6       1
                                SA1       1
                                NZ        X1,*
                                SA1       CBP
                                LX1       59
                                NG        X1,**+3
                                SX6       220314B
                                LX6       42
                                SA6       1
                                JP        **-4
                          *     NO
                                ENDM
000000  0000000000000000000    CBP       BSS       1
000001  0000000000000000000    FIRST     BSS       1
000002  0000000000000000000    IN        BSS       1
000003  0000000000000000000    OUT       BSS       1
000004  0000000000000000000    LIMIT     BSS       1
                                .
000005  1116202524000000000    INPUT     VFD       D30/INPUT
000006  1725242025240000000    OUTPUT    VFD       D36/OUTPUT
                                MACRO     WRITE,FIRSTWA,LASTWA
                                REVEAL    OUTPUT,248,FIRSTWA,LASTWA,FIRSTWA,LASTWA+1

                                ENDM                                                  FIN13001
                                .                                                     FINAL002
                                . FINAL IS A MACRO WHICH TERMINATES A PROGRAM         FINAL002
                                .                                                     FINAL003
                                MACRO FINAL                                           FINAL004
                                SA1 EXIT                                              FINAL005
                                BX6 X1                                                FINAL006
                                SA6 1                                                 FINAL007
                                PS                                                    FINAL008
                                ENDM                                                  FINAL009
000007  0516040000000000000    EXIT      VFD D18/END                                 FINAL010
                                .
                                TEST      WRITE     MESS1,MESS2
```

Figure 4. ASCENT Verification Program

```
000010  7160000112            R TEST     SX6 MESS1
                  5160000001  R          SA6 FIRST
000011  7160000120            R          SX6 MESS2
                  5160000002  R          SA6 IN
000012  7160000112            R          SX6 MESS1
                  5160000003  R          SA6 OUT
000013  7160000121            R          SX6 MESS2+1
                  5160000004  R          SA6 LIMIT
000014  5150000006            R          SA5 OUTPUT
                  7160000024             SX6 24B
000015  12656                            BX6 X5+X6
            5160000000        R          SA6 CBP
000016  7150031117                       SX5 31117B
                  20552                  LX5 52B
000017  7160000000            R          SX6 CBP
                  12656                  BX6 X5+X6
000020  5160000001                       SA6 1
000021  5110000001              *        SA1 1
                  0311000021  R          NZ X1,*
000022  5110000000            R          SA1 CBP
                  20173                  LX1 73B
000023  0331000026            R          NG X1,**3
                  7160220314             SX6 220314B
000024  20652                           LX6 52B
            5160000001                   SA6 1
000025  0200000021            R          JP *-4
000026  46000                  +          NO
                                        FINAL
            5110000007        R          SA1 EXIT
                     10610               BX6 X1
000027  5160000001                       SA6 1
                  0000000000             PS

                                        ORG       **+50
000112  3401230305162455011G   MESS1     DPC       501ASCENT AND LOADER SEEM TO WORK
000113  0455141701040522552G
000114  0505155524175527172G
000115  1355555555555555555G
000116  5555555555555555555G
000117  0000000000000000000G              CON       0
000120  0000000000000000000G   MESS2     CON       0
                                        END       TEST
```

NUMBER OF LINES WITH DIAGNOSTICS --- 0

```
CORE MAP  00.02.02.  NORMAL                CONTROL                                                    000100    000221    000000    000000
          --TIME---LOAD MODE --L1--L2-----TYPE-------------USER-------CALL---------------FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
          FWA LOADER  045123  FWA TABLES  045111
          -PROGRAM----ADDRESS-                    --LABELED---COMMON--
          TEST        000100
          --ENTRY-----ADDRESS-                                REFERENCES
          TEST        000110
          ---UNSATISFIED EXTERNALS-----                       REFERENCES
```

ASCENT AND LOADER SEEM TO WORK

```
00.01,52. TEST002. READ.
00.01,55. TEST002. PP 003 SEC.
00.01,56. TEST002. TEST,17,100,50000.
00.01,56. TEST002. COMMENT, ASSEMBLE AND EXECUTE TO
00.01,56. TEST002. COMMENT, VERIFY THAT ASCENT IS
00.01,57. TEST002. ASCENT,L,LGO,PROG.
00.02,01. TEST002. PROG.
00.02,03. TEST002. CP  000.638 SEC.
00.02,03. TEST002. PP  005.543 SEC.
SCOPE OPERATING SYSTEM - VERSION 2.0, JULY 1966

          CMR            ED  2

          Z        4015      0000
          Z        4016      0000
          Z        4017      0000

NUMBER OF LINES WITH DIAGNOSTICS  ---
```

```
REWIND(MAGTAPE)
1,,INPUT
 WEOF(MAGTAPE)
1,6,INPUT
WEOF(MAGTAPE)
WEOF(MAGTAPE)
REWIND(MAGTAPE)
```

— Page 1 of printout —

```
COPYN TEXT CARDS
                          SKIPF(MAGTAPE,1)
                          SKIPR(MAGTAPE,4)
                           REC3,,MAGTAPE
                          SKIPR(MAGTAPE,2)
                          1,,MAGTAPE
                          SKIPR(DISC,-1)
```

— Page 2 of printout —

```
ASCENT - VERSION 2.0                    PAGE NO,      1
```

Figure 5. COPYN Verification Program

```
ED  0                          ASCENT - VERSION 2.0                    PAGE NO.     2

********************************* WARNING - NO ENTRY POINTS     ********************,*************
                                ASCENT REC6                                              00001
                               ,THE COPYN ROUTINE PERFORMED CORRECTLY A TEST WHICH       00002
                               ,INCLUDED REWIND FILE, WRITE END OF FILE, SEARCH FILE BY  00003
                               ,NAME AND NUMBER, AND COPY A RECORD,                      00004
                                END                                                      00005

NUMBER OF LINES WITH DIAGNOSTICS ---          0
```

Page 4 of printout

```
15.50.59. COPY011. READ.
15.51.01. COPY011. PP  002 SEC.
15.51.02. COPY011. COPYNT,10,100,50000.                    2
15.51.02. COPY011. 42,MCMURRAY,4M664
15.51.02. COPY011. COMMENT,PROGRAM COPYNT TESTS COPYN AND
15.51.02. COPY011. COMMENT,OUTPUTS A MESSAGE TO PRINTER
15.51.03. COPY011. COMMENT,GIVING STATUS OF COPYN
15.51.17. COPY011. REQUEST MAGTAPE.
15.51.17. COPY011.  (51 ASSIGNED)
15.51.18. COPY011. COPYN(0,MAGTAPE,INPUT)
15.51.20. COPY011. COPYN(0,DISC,MAGTAPE)
15.51.25. COPY011. ASCENT(LIST,CI,DISC)
15.51.26. COPY011. CP  001.104 SEC.
15.51.26. COPY011. PP  007.651 SEC.
SCOPE OPERATING SYSTEM - VERSION 2.0, JULY 1966
```

Page 5 of printout

```
                 PROGRAM FNVALID (TAPE 1)
000003           DOUBLE PRECISION X
000003           ITAPE=1
000004           WRITE(ITAPE,100)
000010    100    FORMAT(* FORTRAN VALIDATION *)
000010           X=DSQRT(100.0D+000)
000013           IF(DABS(X-10.0D).LE..0000001D)GO TO 1
000026           WRITE(ITAPE,101)X
000034    101    FORMAT(* ERROR.TEST FAILED, X=*, D30.11)
000034           CALL EXIT
000035    1      WRITE(ITAPE,102)
000041    102    FORMAT(* TEST SUCCESSFUL *)
000041           END
```

Figure 6. FORTRAN Verification Program

```
PROGRAM LENGTH INCLUDING I/O BUFFERS
002117

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1       -  000035    100     -  000050    101     -  000062    102     -  000067

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
ITAPE   -  000105    X       -  000103

START OF CONSTANTS
000043

START OF TEMPORARIES
000073

START OF INDIRECTS
000103

UNUSED COMPILER SPACE
013300
```

```
CORE MAP  00.00.40.  NORMAL                CONTROL                                                    000100    004721    000000    000000
          ---TIME---LOAD MODE --L1--L2-----TYPE--------------USER---+---CALL-----------FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
          FWA LOADER  042123  FWA TABLES  041757
          -PROGRAM----ADDRESS-                    --LABELED---COMMON--
          FNVALID     000100
          SYSTEM      002217
          OUTPTC      003132
          DSQRT       003526
          DABS        003603
          GETBA       003620
          KODER       003637
          XRCL        004714
          --ENTRY-----ADDRESS-                                          REFERENCES
          FNVALID     000101
          Q8NTRY      002220            FNVALID     000102

          SYSTEM      002365            OUTPTC      003146
                                        DSQRT       003564
                                        DABS        003612
                                        KODER       004530

          SYSTEMC     002332
          SYSTEMP     002360
          END         002257            FNVALID     000142

          STOP        002306
          EXIT        002300            FNVALID     000134

          ABNORML     002315            OUTPTC      003147
                                        KODER       004531

          OUTPTC      003134            FNVALID     000106    000107    000130    000132    000133    000137    000140

          DSQRT       003527            FNVALID     000111

          DABS        003604            FNVALID     000122

          GETBA       003620            OUTPTC      003142

          KODER       003640            OUTPTC      003136    003161

          XRCL        004714            OUTPTC      003153    003204    003214    003217    003242

          ----UNSATISFIED EXTERNALS-----                                REFERENCES

FORTRAN VALIDATION
TEST SUCCESSFUL
```

```
00.00.34. FNVA000. READ.
00.00.34. FNVA000. PP 000 SEC.
00.00.34. FNVA000. FNVALID,12,777,45000.
00.00.35. FNVA000. RUN(S)
00.00.39. FNVA000. LGO(OUTPUT)
00.00.43. FNVA000. END    FNVALID
00.00.43. FNVA000. CP  000.201 SEC.
00.00.43. FNVA000. PP  002.841 SEC.
```

PERT/TIME
VERSION 1.0

6400/6600 PERT/TIME operating under SCOPE Version 2.0 has now been released. The release consists of:

A tape containing a compiled absolute binary file, two end-of-files, and the source file of the program

A sample test deck consisting of two data decks type "A" and type "B" input

Two verification decks

PERT/TIME under SCOPE Version 2.0 differs from PERT/TIME under SCOPE Version 1.1 internally only. Overlays have been implemented and chains removed. When requesting PERT/TIME Version 1.0, the operating system under which it is to operate must be specified.

The following document is available for 6000 PERT/TIME:

64/6600 PERT/TIME Reference Manual, Pub. No. 60133600

INSTALLATION INSTRUCTIONS

To load the absolute binary overlay program from tape to F_N PERT66, common PERT66, and execute data, the following control and data cards are required.

```
PERT, 1, 1000, 150000.
REQUEST TAPE 5.
REWIND (TAPE 5)
COPYBF (TAPE 5, PERT66)
COMMON PERT66.
REWIND (PERT66)
PERT66.
R/S
PERT network
R/S
EOF
```

R/S=record separator 7-8-9 in column 1

EOF=end-of-file 6-7-8-9 in column 1

To execute additional networks, the following control cards are required:

```
PERT 2, 1, 1000, 150000.
COMMON PERT66.
REWIND (PERT66)
*
PERT66.
**
R/S
PERT network
EOF
```

*If tapes are required, insert as follows:

```
REQUEST TAPE 4.                    To input old master tape
REQUEST TAPE 6.                    To make and save new master file
```

**If TAPE 4 and TAPE 6 are used, these cards should follow PERT66. also.

```
REWIND (TAPE 4)
REWIND (TAPE 6)
```

To unload and save tapes the cards required are:

```
UNLOAD (TAPE 4)
UNLOAD (TAPE 6)
```

following the rewinds.

To list or punch the PERT source file, the following control cards are required:

```
PERT, 1, 200, 70000.
REQUEST TAPE.
REWIND (TAPE)
COPYBF (TAPE, X, 3)
COPYBF (TAPE, OPERATOR)
REWIND (TAPE)
R/S
EOF
OPERATOR=PUNCH for punched output
         =OUTPUT for listing
```

To execute or compile the source file, the following control cards are required:

```
PERT, 1, 500, 70000.
REQUEST TAPE.
REWIND (TAPE)
COPYBF (TAPE, XX, 3)
COPYBF (TAPE, TAPE 5)
REWIND (TAPE 5)
RUN (S, 150000,, TAPE 5,,, 70000)
*
LOAD (LGO)
```

```
NOGO.
**
PERT66.
R/S
PERT network
EOF
```

*To save load-and-go file insert:

```
REQUEST TAPE 8.
REWIND (LGO)
COPYBF (LGO, TAPE 8)
REWIND (LGO)
```

**To save overlay tape insert:

```
REQUEST TAPE 9.
REWIND (PERT66)
COPYBF (PERT66, TAPE 9)
REWIND (PERT66)
```

For larger test cases of 1000 activities or more, CPU time may be saved by assigning TAPE 1, TAPE 2, and TAPE 3 to magnetic tapes.

CLASSIFICATION

PERT/TIME
ACTIVITY REPORT

REPORTING ORGN. CONTRACT NO. 333

PERT TEST(VARIOUS SORT KEYS) N
1ST SORT KEY PREDECESSOR EVENT NO.
2ND SORT KEY SUCCESSOR EVENT NO.
3RD SORT KEY LEAST SLACK
4TH SORT KEY EXPECTED DATE (TE)

TERM-
REPORT DATE- 9/17/64
RELEASE DATE- 9/17/64

	EVENT PRED.	EVENT SUCC.	ACTIVITY DESCRIPTION	PROB.	ACTIV. TIME	DATE EXPECTED	DATE ALLOWED	DATE COMP/SCHED	REMAINING SLACK	REMAINING TIME	ORG.	ACCOUNT NO.
	00	01		0	0.0		9/30/64	A 9/17/64	1.7	0.0	ORG1	012345678900
I	01	02		.99	7.6	11/10/64	11/23/64		1.7	7.6	ORG4	01
I	01	03		.99	3.6	10/13/64	11/24/64		5.9	3.6	ORG3	001257567890
I	01	04		.99	5.3	10/26/64	11/19/64		3.7	5.3	ORG1	51
	02	05		.88	4.3	12/11/64	12/23/64		1.7	11.9	ORG4	02
	02	09		.99	4.3	12/11/64	4/28/65		19.2	11.9	ORG1	31
	03	05		.99	4.1	11/11/64	12/23/64		5.9	7.7	ORG4	03
	03	06		.99	4.1	11/11/64	12/30/64		6.6	7.7	ORG1	01
	04	07		.99	2.3	11/10/64	12/ 8/64		3.7	7.6	ORG2	53
	04	08		.99	3.8	11/20/64	1/18/65		7.7	9.1	ORG1	52
	05	10		.86	4.0	1/12/65	1/22/65		1.7	15.9	ORG5	023456718560
	06	10		.99	3.0	12/ 3/64	1/22/65		6.9	10.7	ORG1	03
	06	11		.99	3.0	12/ 3/64	1/21/65		6.6	10.7	ORG2	81
	07	11		.99	6.0	12/23/64	1/21/65		3.7	13.6	ORG2	52
	08	12		.99	4.3	12/22/64	2/17/65		7.7	13.4	ORG1	54
	09	13		.99	0.0	12/11/64	4/28/65		19.2	11.9	ORG1	31
	10	15		.85	5.0	2/16/65	2/26/65		1.7	20.9	ORG2	01
	11	16		.99	3.5	1/20/65	2/15/65		3.7	17.1	ORG2	81
	12	14		.99	3.1	1/15/65	3/10/65		7.7	16.5	ORG5	51
	13	18		.99	1.1	12/18/64	5/ 5/65		19.2	13.0	ORG4	71
	14	17		.99	4.6	2/17/65	4/12/65		7.7	21.1	ORG2	41
	15	19		.84	2.8	3/ 8/65	3/18/65		1.7	23.7	ORG5	61
	15	22		.99	6.8	4/ 5/65	6/17/65		10.6	27.7	ORG1	71
	16	20		.99	4.6	2/22/65	3/18/65		3.7	21.7	ORG2	82
	17	21		.99	2.5	3/ 5/65	4/29/65		7.7	23.6	ORG1	42
	18	22		.99	6.1	2/ 3/65	6/17/65		19.2	19.1	ORGJ	72
	19	20		.84	0.0	3/ 8/65	3/18/65		1.7	23.7	ORG2	82
	20	22		.99	7.6	4/29/65	6/17/65		7.0	31.3	ORG4	71
	20	25		.80	14.5	6/16/65	6/29/65		1.7	38.2	ORG1	11
	20	27		.99	13.6	6/10/65	8/ 5/65		8.1	37.3	ORG4	21
	20	30		.99	8.6	5/ 6/65	9/24/65		20.0	32.3	ORG2	91
	21	24		.99	3.0	3/26/65	5/20/65		7.7	26.6	ORG2	41
	22	23		.99	-0.0	4/29/65	6/17/65		7.0	31.3	ORG2	72
	23	26		.99	4.1	5/27/65	7/15/65		7.0	35.4	ORG2	22
	23	27		.99	5.1	6/ 3/65	8/ 5/65		9.0	36.4	ORG1	21
	24	25		.99	5.6	5/ 5/65	6/29/65		7.7	32.2	ORG5	42
	25	27		.78	5.5	7/26/65	8/ 5/65		1.7	43.7	ORG1	11
	25	29		.99	2.0	6/30/65	8/19/65		7.1	40.2	ORG4	12
	26	27		.99	3.0	6/17/65	8/ 5/65		7.0	38.4	ORG5	23
	26	28		.99	3.1	6/18/65	8/20/65		9.0	38.5	ORG1	22
	27	28		.78	2.1	8/ 9/65	8/20/65		1.7	45.8	ORG2	23

CLASSIFICATION

Figure 7. Partial Sample of Type A PERT/TIME Verification Deck

CLASSIFICATION

PAGE 1

PERT/TIME
MILESTONE REPORT

REPORTING ORGN.

CONTRACT NO.
333

TERM

REPORT DATE- 9/17/64
RELEASE DATE 9/17/64

PERT TEST(VARIOUS SORT KEYS) N

LEVEL/SUMMARY ITEM 5/ 0

S=SCHEDULE COMPLETION DATE
E=EARLIEST COMPLETION DATE
L=LATEST COMPLETION DATE

A=ACTUAL COMPLETION DATE
M=EARLIEST COMPLETION DATE
AFTER MGMT ACTION

MILE-STONE CODE	EVENT NO.	EVENT DESCRIPTION	SLACK	DATE	P YR 1964 JFMAMJJASOND	1965 JFMAMJJASOND	66 67 68 69 L YR
14	14	EVENT 14	7.7	1/18/65		S	
				1/15/65		E	
				3/10/65		L	
15	15	EVENT 15	1.7	2/12/65		S	
				2/16/65		E	
				2/26/65		L	
16	16	EVENT 16	3.7	1/22/65		S	
				1/20/65		E	
				2/15/65		L	
18	18	EVENT 18	19.2	1/11/65		S	
				12/18/64	E		
				5/ 5/65		L	
19	19	EVENT 19	1.7	3/ 5/65		S	
				3/ 8/65		E	
				3/18/65		L	
20	20	EVENT 20	1.7	3/ 5/65		S	
				3/ 8/65		E	
				3/18/65		L	
21	21	EVENT 21	7.7	3/15/65		S	
				3/ 5/65		E	
				4/29/65		L	
22	22	EVENT 22	7.0	5/ 3/65		S	
				4/29/65		E	
				6/17/65		L	
23	23	EVENT 23	7.0	5/ 3/65		S	
				4/29/65		E	
				6/17/65		L	
24	24	EVENT 24	7.7	4/ 6/65		S	
				3/26/65		E	
				5/20/65		L	
25	25	EVENT 25	1.7	6/18/65		S	
				6/16/65		E	
				6/29/65		L	

Figure 8. Partial Sample of Type B PERT/TIME Verification Deck

SCOPE
VERSION 2.0

NEW FEATURES AND MODIFICATIONS

64/6600 SCOPE Version 2.0 contains a powerful relocatable loader which allows subprograms to be assembled or compiled independently and then brought together prior to execution in one of three fashions: normal loading, segmentation, overlays. A new library utility routine, 64/6600 COPYN, has been provided to aid in program file updating. The utility routine, CATALOG, has been modified to allow for the new binary formats. SCOPE Version 2.0, including COPYN and CATALOG, is described in the SCOPE Reference Manual, Pub. No. 60173800.

DMP

1. The SCOPE 2.0 version of DMP reformats the Exchange package to label the P, RA, EM, and FL parts, as well as the individual A, B, and X registers.

2. DMP eliminates the last n - 1 words of an n-word block of identical words in central memory. It also eliminates the last n - 1 words of an n-word block of the form yyyyyy 60000000000000yyyyy in central memory. In this case, $(Y)_{17-0}$ is compared to Y for identity. If they are not identical, no elimination takes place.

3. Up to four words per line are printed. Less than four words will be printed under the following conditions:

   a. If the initial address (first dump argument) is not divisible by four, the line is truncated in such a way that the next line begins with a word whose address is divisible by 4.

   b. If there are n-word blocks of the sort described above, either that line or the next one is truncated to make the addresses in the left column divisible by 4. The line on which this occurs is a function of the addresses which occur after deletions and the number of renumberings which may take place on the same line. Addresses are printed down the left column only, unless there have been deletions on the line. In this case, the address of the first word after the deleted block is printed to flag the presence of a deletion. In columns 2, 3, and 4 the address is separated by a special character (Display Code 65) which prints as a blank if the printer drivers in use have not been modified for a 64-character set.

4. DMP provides an automatic dump of the area around a stop location if the exchange dump is requested. If P >77 then (RA + P - 77) through (RA + P + 77) is dumped. If P = 0 and RA >77 then (RA - 77) through (RA + 77) is dumped. In all other cases (RA + 0) through (RA + 100) is dumped.

5. The entire control point area of the requesting program is dumped if a card of the form

   DMP(n,n) $n \neq 0$

   is encountered. The label reads DMPC.

6. Absolute core dumps are produced by a

   DMP(4xxxxx,4yyyyy)

   where xxxxx defines the lower bound and yyyyy defines the upper bound of the absolute core locations wanted. For example:

   DMP(400000,413777)

   dumps the entire central resident (0 to 13777). Label reads DMPA.

7. The contents of A1 through A7 are printed on the same line with the corresponding A register.

8. The output buffer is emptied before the DMP output is produced. DMPs will not be lost if the output file is busy.

9. This version works with up to 5 disks. It has been checked out on a 1 disk machine.

Catalog

Catalog is a 2.0 library routine that accepts any file in the format described below as input and gives a listing of the packages contained in that file. The listing includes the length of each logical record, the names of the packages contained in each record, the length of each of these packages, and a checksum of each package.

Packages input to catalog must be divided into tables with header words conforming to one of the following four descriptions:

1. The first table has a control number (CN) of 7700 and the second table has a CN of 3400 (Standard format for ASCENT deck).
2. The first table has a CN of 7700 but the second does not have 3400 (ASPER deck).
3. The first table has a CN of 3400 (ASCENT deck with 77 table missing).
4. The first table has neither 7700 nor 3400 (ASPER deck with first table missing).

The output listing contains the following five sorts of information:

1. RECORD — The number of the logical record with respect to its position on the tape. Zero-length records produce a record number.
2. LENGTH — The entire length of the logical record, including all 77 tables of the packages on the record.
3. PACKAGE — The name of the package found at a well-defined location within the file. Any name beginning with a character which is nonalphanumeric or blank or zero is illegal and a minus sign (-) will replace the name in the listing. When this occurs no package length or checksum will appear.
4. CHKSUM — Computed by adding together all the words of the package and along with each word adding a counter that is decremented each time a new word is added in. This insures a unique checksum in the event that the program gets out of sequence. The final answer is then folded into 12 bits.
5. LENGTH — The length of the package. LENGTH contains all the words except those in a 77 table, if one appears.

Catalog input/output is accomplished using the Circular Buffer I/O routine.

To call the routine use: CATALOG (file1, file2). Information is taken from file1 and listed on file2. If the parameters are omitted, LIBRARY and OUTPUT are assumed.

COPYCR

COPYCR has been changed to correspond as closely as possible to the concept of a coded record. Internally, a coded record is a string of display coded characters terminated by a zero byte. It is usually produced by reading a card or preparing a line image destined for the printer or a display. However, coded records may be grouped together into a binary record.

COPYCR copies the requested number of the next available coded records. If the copy is from disk it reads binary records until it has copied the requested number of coded records, leaving the file positioned at the binary record following that which contained the last coded record. If the copy is from coded tape the requested number of records are copied but the tape may be left positioned beyond these records, depending on buffer size.

Example:

```
JOB, 10,1000,40000.
COPYCR (INPUT, OUTPUT, 3)
COPYCR (INPUT, OUTPUT, 1)
7, 8, 9
CARD1
CARD2
CARD3
CARD4
7, 8, 9
CARD5
6, 7, 8, 9
```

Produces on output

```
CARD1
CARD2
CARD3
CARD5
```

Note that CARD4 in this example is not output because it is contained in the binary record read by the first copy.

LIMITATIONS AND KNOWN DEFICIENCIES

1. All PSR's have been corrected through PSR 72 except for 61, 65, 66, 67, 71.

2. When a user call contains a list of segments and single program names to be loaded, and the program names follow the segment names, the single programs are not loaded by the PPU routine LDR. LOADER produces the message "REQUESTED SEGMENT INCOMPLETE" and a fatal error flag is returned. Thus, when a segment is composed of more than a single named segment (which should be unusual since the structure permits the formation of a segment with sections and programs), the user must place all single program names in the SL list before the segment name.

3. When a user call requests a named segment or section containing a program of which there is more than one copy on the requested file, or when such a program is named in the SL list along with other programs, it is then possible for more than one copy of the program to be loaded into core. This happens if the file is positioned in such a manner that more than one copy of the program is encountered before the named segment, section or list of programs has been completely loaded from the file. Please note that the order of loading from an SL list is: named segment, named section, individual programs. Further, the load of each named segment or named section is treated independently.

   If there is a danger of the above situation occurring, the user should assign the program name to a separate section or segment to isolate the search for it from other program loading. If this is not observed, and more than one copy of a program is loaded accidentally, the first copy is linked to all programs at the same or lower level, while the last copy loaded is linked to all higher level segments.

4. COPYN gets into a loop if the current input file is positioned at end of information and a record specified (p_1 or p_2) on the record identification card is either a non-existent record or record one of the file.

5. When in segment mode, labeled COMMON is local to the segment that declared it and can not be used as universal storage. This is not explicitly stated in the reference manual.

6. It is the user's responsibility to change the size of the FNT when necessary. If a large number of programs are being run at once and the size of the FNT is not reset appropriately, the FNT will fill up. No recovery is then possible.

7. When no file name is given on a BKSP card, a preset file name and number are used, and that file backspaced. This does not affect the user's program.

ASCENT
VERSION 2.0

NEW FEATURES AND MODIFICATIONS

Version 2.0 of ASCENT is described in the ASCENT ADB, Pub. No. 60175400.

Conversion of ASCENT Programs

On Page 5-11 in the Chippewa Operating System Reference Manual (Publication No. 60134400) rules are given for coding a subroutine in such a way as to define the relocatable parts, and the point at which execution is to begin. These rules are also valid for coding a single independent program in ASCENT 1.1. They provide that the first two words of the assembled routine should be:

```
VFD     D24/NAME,N18/0,A18/end
VFD     A18/reloc,A18/end,N24/params
```

(Note that "D24" and "N18" in the first word are based on the assumption that the name of the program has four characters; "NAME" is used in the example.)

"params" is an integer defining the number of locations that are to be left vacant by the loader before the first instruction; at load time the parameters from the control card are normally loaded into successive locations beginning at RA+2; the space for these parameters ends at RA+params+1; and execution will begin at the first instruction, in location RA+params+3.

All the instructions in the routine, i.e., the words whose addresses may be subject to relocation, lie in the area from RA+params+3 through reloc-1. "reloc" is the address of the first constant, i.e., the first word that must not be relocated, and the last constant must be at end-1.

These rules do not hold for SCOPE 2.0. In order to convert a program coded in ASCENT machine language for Chippewa 1.1 into a program for SCOPE 2.0, remove the two VFD cards described above, the "params EQU n" card, (which is now unnecessary) the "SUBRT" card, if any (as this is not a valid pseudo-op in ASCENT 2.0), the "reloc EQU **1+1" card, and the "end EQU **+1" card, if any (as it is no longer necessary to define the areas respectively occupied by instructions and constants). Immediately after the ASCENT card at the beginning of the program, insert the card:

```
ENTRY   start
```

where "start" is the symbol in the location field of the instruction at which execution is to begin. Every program in ASCENT 2.0 must have at least one entry point. Execution of a program can only begin at an entry point.

Replace the END card that terminates the program with the card:

```
END     start
```

where "start" is the name of the entry point at which execution is to begin. This is the simplest way of specifying the point for beginning execution of a single independent program.

Both under Chippewa 1.1 and under SCOPE 2.0, the parameters from the control card are to be found in locations RA+2, RA+3, etc. If the program addresses these by number, or by symbolic constants that are equated to 2, 3, etc., no further change has to be made. But if they are addressed by symbols that have been defined through BSS, BSSZ, or CON cards at the beginning of the program, so as to occupy space in the area whose length was defined by params, then one of the following two changes should be made:

1. Remove those defining cards, and replace them with EQU cards that directly equate the symbols to integers 2, 3, etc.
2. Leave the defining cards in the program; immediately before the first one insert the card:

   ```
   ORG     2
   ```

   and immediately after the last one insert the card:

   ```
   ORG     *
   ```

Suppose the following is a program for Chippewa 1.1:

```
          ASCENT    TRYME
PARAMS    EQU       5
          VFD       D30/TRYME,N12/0,A18/TERMIN
          VFD       A18/RELIC,A18/TERMIN,N24/PARAMS
PARAM1    BSS       1
PARAM2    BSS       1
PARAM3    BSS       4
GOMAN     SA1       CONA
          SA2       CONB
          FX6       X1*X2
          SA3       PARAM2
          ZR        X3,PUT
          FX6       X1/X2
PUT       SA6       CONA
          PS
RELIC     EQU       **1+1
CONA      CON       1.
CONB      CON       2.
TERMIN    EQU       **1+1
          END
```

For SCOPE 2.0, this should be changed to one of the following:

Example 1:

```
            ASCENT   TRYME
            ENTRY    GOMAN
PARAM1      EQU      2
PARAM2      EQU      3
PARAM3      EQU      4
GOMAN       SA1      CONA
            SA2      CONB
            FX6      X1*X2
            SA3      PARAM2
            ZR       X3,PUT
            FX6      X1/X2
PUT         SA6      CONA
            PS
CONA        CON      1.
CONB        CON      2.
            END      GOMAN
```

Example 2:

```
            ASCENT   TRYME
            ENTRY    GOMAN
            ORG      2
PARAM1      BSS      1
PARAM2      BSS      1
PARAM3      BSS      4
            ORG      *
GOMAN       SA1      CONA
            SA2      CONB
            FX6      X1*X2
            SA3      PARAM2
            ZR       X3,PUT
            FX6      X1/X2
```

```
PUT     SA6     CONA
        PS
CONA    CON     1.
CONB    CON     2.
        END     GOMAN
```

In ASCENT 1.1, the pseudo-op ORG had no function in ASCENT programs (though it did work in ASPER programs), because a program would ordinarily be loaded starting at location RA+0.

In ASCENT 2.0, ORG does have a function, but previous ASCENT programs can be successfully modified without using ORG. In the absence of an ORG card, ASCENT 2.0 assumes that the program begins with:

```
        ORG     *
```

which means, "assemble the following for loading into the lowest-numbered available section of relocatable storage;" initially, that is, into relocatable 0. So the program will be assembled with addresses beginning at 0, and at load time it will be loaded into locations beginning at RA+100_8; all the relocatable addresses in the program will be modified accordingly. The parameters from the control card will be stored at run time, as before, starting at location RA+2, so there is no need for the user's program to explicitly reserve space for them.

In the second suggested version of the program for SCOPE 2.0, the

```
        ORG     2
```

card causes everything between it and the next ORG card to be loaded into locations beginning at RA+2; thus PARAM1, PARAM2, and PARAM3 are defined correctly. The program itself must be preceded by

```
        ORG     *
```

so that it will be assembled as relocatable code, for eventual loading in an area beginning at RA+100_8.

When two or more programs are to be assembled separately, but are expected to be loaded together at some time, they can be coded according to either of the suggested models given above. They can both find the parameter string starting at location RA+2. The first program will be loaded at RA+100_8, but the others will be loaded higher up in memory.

LIMITATIONS AND KNOWN DEFICIENCIES

1. All PSR's through PSR 19 have been corrected except PSR 18.

2. Macros do not pass the names of parameters that are themselves other macro names.

3. DPC instructions do not expand correctly within macros.

4. The use of literal names results in UU diagnostics at the END card.

FORTRAN
VERSION 2.0

NEW FEATURES AND MODIFICATIONS

Version 2.0 of 64/6600 FORTRAN is an improvement of Version 1.1. Some changes were necessary to allow the system to operate under the 2.0 SCOPE relocatable operating system. Many new features have been added and deficiencies present in 64/6600 FORTRAN Version 1.1 have been corrected. The following is a list of changes made to the system. For a more thorough description of the additions to the system see the Conversion Guide, FORTRAN Version 1.1 to 2.0, Pub. No. 60175500.

1. The compiler now compiles subprograms independently, and produces a relocatable record on a specified file. The 2.0 SCOPE loader now takes care of loading and linking subprograms together.

2. The compiler no longer recognizes a SEGMENT card nor does it interpret CALL CHAIN as a special library call. The chaining available in Version 1.1 has been replaced by the more versatile OVERLAY and SEGMENTATION capabilities of the 2.0 operating system.

3. The compiler now recognizes overlay and segment control cards if they appear between subprograms and if they begin after column six. When finding such a control card, the compiler lists it and transfers it to the binary output file(s). This is done to aid the programmer in overlay and segment preparation.

4. The format of the beginning of each subprogram has been modified and the zero words previously saved for every subroutine argument are now only saved for each argument after the sixth. The format of subprograms is as follows.

   [Zero words for each argument after the sixth]

   [NAME NN]

   [ENTRY/EXIT LINE]

   Routines written in machine language should be in this format as the error traceback routine (SYSTEM) which has been implemented depends on this.

5. The FORTRAN compiler is called by the control card:

   RUN (cm,fl,bl,if,of,rf,lc,as,cs)

   cm compiler mode option; (if omitted, assume G; if unrecognized, assume S)

   - G compile and execute with no source list, unless explicit LIST cards appear in the deck or unless errors are present in the source deck
   - S compile with source list, no execute
   - P compile with source list and punch deck on file PUNCHB, no execute
   - L compile with source and object code list, no execute
   - M compile with source and object code list, produce a punch deck on file PUNCHB, no execute

fl object program field length (octal); if omitted, it is set equal to the field length at compile time.

bl object program I/O buffer lengths (octal); if omitted, assumed to be 2011B

if file name for compiler input; if omitted assumed to be INPUT

of file name for compiler output; if omitted, assumed to be OUTPUT

rf file name on which the binary information is always written; if omitted, assumed to be LGO

lc line-limit (octal) on the OUTPUT file of an object program. If omitted, assumed to be 10000_8.

as ASA switch; non-blank selects option.

cs cross reference; non-blank selects option.

6. The storage necessary for I/O buffers is now made part of the PROGRAM.

7. The operational characteristics of the compiler have been slightly modified to have more meaning under a relocatable system.

   - The length of each subprogram is written in the output file.
   - The unused compiler space for each subprogram is written in the output file.
   - The name and length of each common block is placed in the output file.
   - When the variable map is produced, if the variable is in common, the address given is relative to the start of the common block. Therefore after the address a "C" along with the octal ordinal of the common block, under block assignments, is given.
   - When the compiler has processed all input, the total number of errors detected during that compilation process is placed in the Dayfile.
   - When fatal errors are detected in a subprogram, no binary output for that subprogram is produced and no variable map is written. If the compilation mode was "G", the program will not be automatically executed.

8. The compiler is now sectioned into three overlays, (0,0), (1,0), (1,1). The (0,0) overlay, whose entry point is RUN, contains the code necessary to terminate all output buffers at the end of compilation and the code necessary to transfer to both level (1,0) of the RUN compiler and to level (1,0) of the ASCENT assembler. Level (1,0) is the main body of the compiler.

   Level (1,1) of the FORTRAN compiler is called when it becomes necessary to list full line diagnostics.

9. The compiler transfers control to the ASCENT assembly system when an ASCENT or ASPER header card is detected. This provides the programmer with easy linkage to and from a powerful assembly system. When the assembler completes its processing, control returns to level (0,0) of FORTRAN. If no more input is present, level (0,0) terminates the compilation. Otherwise, level (1,0) is reloaded and the compilation process continues. The assembly routines included on the Version 1.1 compiler (ASCENTF, MACHINE) are no longer part of the RUN compiler.

10. Non fatal diagnostics have been implemented. Each error results in a two or three letter diagnostic listed at the point of error detection. All two letter codes are non fatal while all three letter codes (which usually have an F suffixed to them) are fatal. If a listing is requested or if fatal errors are detected, full line diagnostics, indicating the address at which the error occurred are listed at the end of the subprogram.

11. The ENTRY statement has been implemented under the following rules.

    - It cannot appear with the range of a DO.
    - It cannot be labeled.
    - The name may not be followed by a list of arguments as it is assumed to have the same number of arguments as the subprogram in which it occurs.
    - It assumes the same type as the subprogram in which it appears.

12. LIST, NOLIST option has been implemented. If a LIST card, starting after column six appears between subprograms, listing takes place from that point until a NOLIST card, starting after column six is detected. After a NOLIST card is detected no listing takes place until another LIST card appears or a fatal error occurs. If fatal errors occur all subprograms after the NOLIST card will be then listed, as under the G compilation mode. This is due to the extremely costly backspace problem with disk files.

13. Variable format may now be a simply subscripted integer variable.

14. The routine SYSTEM has been expanded to list all diagnostics the object routines require and to provide full error traceback information. Capabilities exist for producing non-standard error recovery and for changing the status of errors from non-fatal to fatal or vice versa.

15. The initialization code formerly compiled when a PROGRAM card was being processed is now in a routine called Q8NTRY. This was done in order to incorporate overlays, replacing the former usage of a SEGMENT header card.

16. Since a certain number of routines are always required at execution time, such as END, Q8NTRY, and SYSTEM, these have all been included as entry points to the routine SYSTEM.

17. All object routines have been modified to call the SYSTEM routine when it becomes necessary to give diagnostics.

18. The I/O routines have been split into several routines so the coder and cracker routines only appear once. G conversion has been implemented. An ASA switch has been implemented which allows for proper ASA format re-scan and ASA P. scaling.

19. Two routines, OVERLAY and SEGMENT, have been included to provide linkage between the FORTRAN and the 2.0 loader. Basically, they translate the FORTRAN call into a recognizable call to the loader.

20. Multiple entry points have been implemented in Version 2.0 so many of the library routines have been combined. Table 1 is a list of the library routines, the entry points they contain, and the external routines they reference. SCOPE Version 2.0 allows library routines to reference other library routines. In order to take advantage of this facility, many of the object time library routines have been reorganized so that all repetitive coding is a separate routine. For example, the BCD format cracker (KRAKER) which was previously contained within both INPUTS and INPUTC is now a separate routine and can be referenced by both of the input routines. Not only have the I/O routines been divided but the mathematical library

Table 1. FORTRAN Library Routine Entry Points

Routine	Entry Points	Externals
ACGOER	ACGOER	SYSTEM, ABNORML
ALNLOG	ALOG, ALOG10	SYSTEM
ASINCOS	ASIN, ACOS	SYSTEM
ATAN	ATAN	SYSTEM
ATAN2	ATAN2	SYSTEM
BACKSP	BACKSP	SYSTEM, ABNORML, GETBA, XRCL
BUFFEI	BUFFEI	SYSTEM, ABNORML, GETBA, XRCL
BUFFEO	BUFFEO	SYSTEM, ABNORML, GETBA, XRCL
CABS	CABS	SYSTEM
CBAIEX	CBAIEX	SYSTEM
CCOS	CCOS	COS, SIN, EXP, SYSTEM
CEXP	CEXP	COS, SIN, EXP, SYSTEM
CLOG	CLOG	ALOG, ATAN2, CABS, SYSTEM
CSIN	CSIN	COS, SIN, EXP, SYSTEM
CSQRT	CSQRT	CABS, SQRT, SYSTEM
DABS	DABS	SYSTEM
DATAN	DATAN, DATAN2	SYSTEM
DBADEX	DBADEX, DBAREX, RBADEX	DLOG, DEXP, SYSTEM
DBAIEX	DBAIEX	SYSTEM
DBLE	DBLE	
DEXP	DEXP	SYSTEM
DISPLA	DISPLA	
DLNLOG	DLOG, DLOG10	SYSTEM
DMOD	DMOD	SYSTEM
DSIGN	DSIGN	SYSTEM
DSINCOS	DSIN, DCOS	SYSTEM
DSQRT	DSQRT	SYSTEM
DUMP	DUMP, PDUMP	OUTPUTC, STOP
DVCHK	DVCHK	
ENDFIL	ENDFIL	SYSTEM, ABNORML, GETBA, XRCL

Routine	Entry Points	Externals
EXP	EXP	SYSTEM
GETBA	GETBA	SYSTEM, ABNORML
IBAIEX	IBAIEX	SYSTEM
IDINT	IDINT	SYSTEM
IFENDF	IFENDF	SYSTEM, ABNORML, GETBA
INPUTB	INPUTB	SYSTEM, ABNORML, GETBA, XRCL
INPUTC	INPUTC	SYSTEM, ABNORML, GETBA, XRCL, KRAKER
INPUTS	INPUTS	SYSTEM, ABNORML, KRAKER
IOCHEK	IOCHEK	SYSTEM, ABNORML, GETBA, XRCL
IOCHEC	IOCHEC	
KODER	KODER	SYSTEM, ABNORML
KRAKER	KRAKER	SYSTEM, ABNORML
LENGTH	LENGTH	SYSTEM, ABNORML, GETBA
LOCF	LOCF, XLOCF	
OUTPTB	OUTPTB	SYSTEM, ABNORML, GETBA, XRCL
OUTPTC	OUTPTC	SYSTEM, ABNORML, GETBA, XRCL, KODER
OUTPTS	OUTPTS	SYSTEM, ABNORML, KODER
OVERFL	OVERFL	
OVERLAY	OVERLAY	LOADER, SYSTEM, ABNORML
PAUSE	PAUSE	
RANF	RANF	
RBAIEX	RBAIEX	SYSTEM
RBAREX	RBAREX	ALOG, EXP, SYSTEM
REMARK	REMARK	
REWINM	REWINM	SYSTEM, ABNORML, GETBA, XRCL
SECOND	SECOND	
SEGMENT	SEGMENT	LOADER, SYSTEM, ABNORML
SINCOS	SIN, COS	SYSTEM
SLITE	SLITE	SYSTEM
SLITET	SLITET	SYSTEM
SNGL	SNGL	

Routine	Entry Points	Externals
SQRT	SQRT	SYSTEM
SSWTCH	SSWTCH	SYSTEM
SYSTEM	SYSTEM,SYSTEMC,SYSTEMP, Q8NTRY, STOP,END,EXIT,ABNORML	
TAN	TAN	SYSTEM
TANH	TANH	EXP,SYSTEM
TIME	TIME	
XRCL	XRCL	

has also been revised. CEXP, the routine which raises a complex number to a power, references the exponential routine and the SINCOS routine both of which may be called individually. Even though more features and diagnostics have been added, significant storage reduction will be noticed if several of the I/O routines are used by a single program.

21. The FORTRAN object routines test for many of the more common cases of incorrect arguments and call the subroutine SYSTEM to handle the error in the standard fashion. The table below lists, for each routine that makes such tests, the condition detected, the standard recovery action (either the answer supplied, or the word "fatal" for fatal errors), and the error number.

Table 2. FORTRAN Object Routine Error Diagnostics

The symbols INF and IND below denote the infinite and indefinite internal words, respectively.

Where an error condition is preceded by "also:" it indicates that the routine in question calls on a subordinate library routine, giving it the arguments indicated, and therefore the subordinate routine may detect some errors of its own and report them under its own error number.

Routine	Condition	Standard Recovery	Error Number
AGGOER	This routine is only called upon detection of a computed or assigned GO TO error.	Fatal	1
ACOS (R)	R = INF or R = IND or abs (R) .GT. 1.0	+IND +IND	2
ALOG (R)	R = INF or R = IND or R .LT. 0 R = 0	+IND -INF	3
ALOG10 (R)	R = INF or R = IND or R .LT. 0 R = 0	+IND -INF	4
ASIN (R)	R = INF or R = IND or abs (R) .GT. 1.0	+IND	5

Routine	Condition	Standard Recovery	Error Number
ATAN (R)	R = INF or R = IND	+IND	6
ATAN2 (R1, R2)	(R1 or R2) = (INF or IND) R1 = R2 = 0	+IND +IND	7
CABS (Z)	(real (Z) or imag (Z)) = (INF or IND)	+IND	8
CBAIEX:Z**I	(real (Z) or imag (Z)) = (INF or IND) Z = (0,0) and I .LE. 0	(+IND,+IND) (+IND,+IND)	9
CCOS (Z)	(real (Z) or imag (Z)) = (INF or IND) also: COS (real (Z)) and EXP (imag (Z))	(+IND,+IND)	10
CEXP (Z)	(real (Z) or imag (Z)) = (INF or IND) also: SIN(imag (Z)) and EXP (real (Z))	(+IND,+IND)	11
CLOG (Z)	(real (Z) or imag (Z)) = (INF or IND) also: ALOG (CABS(Z)) and ATAN2 (imag (Z), real (Z))	(+IND,+IND)	12
COS (R)	R = INF or R = IND or abs (R) .GT.1.1E14	+IND	13
CSIN (Z)	(real (Z) or imag (Z)) = (INF or IND) also: SIN(real(Z)) and EXP (imag (Z))	(+IND,+IND)	14
CSQRT (Z)	(real (Z) or imag(Z)) = (INF or IND)	(+IND,+IND)	15
DABS (D)	D = INF D = IND	+INF +IND	16
DATAN (D)	D = INF or D = IND	+IND	17
DATAN2 (D1, D2)	(D1 or D2) = (INF or IND) D1 = D2 = 0	+IND +IND	18
DBADEX: D1**D2	(D1 or D2) = (INF or IND) D1 = 0 and D2 .LE. 0 D1 .LT. 0	+IND +IND +IND	19
DBAIEX: D1**I2	D1 = INF or D1 = IND D1 = 0 and I2 .LE. 0	+IND +IND	20
DBAREX: D1**R2	(D1 or R2) = (INF or IND) D1 = 0 and R2 .LE. 0 D1 .LT. 0	+IND +IND +IND	21
DCOS (D)	D = INF or D = IND or abs (D) .GT.1.1E14	+IND	22
DEXP (D)	D = INF or D = IND D .GT. 741.67	+IND +INF	23
DLOG (D)	D = INF or D = IND or D .LT. 0 D = 0	+IND -INF	24
DLOG10 (D)	D = INF or D = IND or D .LT. 0 D = 0	+IND - INF	25

Routine	Condition	Standard Recovery	Error Number
DMOD (D1,D2)	(D1 or D2) = (INF or IND) D2 = 0 D1 / D2 .GE. 2 ** 96	+IND +IND +IND	26
DSIGN (D1,D2)	D1 = IND or D2 = (0 or INF or IND) D1 = INF	+IND INF with sign of D2	27
DSIN (D)	D = INF or D = IND or abs (D) .GT.1.1E14	+IND	28
DSQRT (D)	D = INF or D = IND or D .LT. 0	+IND	29
EXP (R)	R = INF or R = IND R .GT. 741.67	+IND +INF	30
IBAIEX: I1**I2	I1 = 0 and I2 .LE. 0 I1 ** I2 .GE. 2** 48	0 0	31
IDINT (D)	D = +INF or D = IND or D .GE. 2**59 D = -INF or D .LE. -2**59	2**59-1 1-2**59	32
RBADEX: R1**D2	(R1 or D2) = (INF or IND) R1 = 0 and D2 .LE. 0 R1 .LT. 0	+IND +IND +IND	33
RBAIEX: R1**I2	R1 = INF or R1 = IND R1 = 0 and I2 .LE. 0	+IND +IND	34
RBAREX: R1**R2	(R1 or R2) = (INF or IND) R1 = 0 and R2 .LE. 0 R1 .LT. 0	+IND +IND +IND	35
SIN (R)	R = INF or R = IND or abs (R) .GT.1.1E14	+IND	36
SLITE (I)	I .GT. 6 or I .LT. 0	Proceed	37
SLITET (I1,I2)	I1 .GT. 6 or I1 .LE. 0	I2 =2	38
SQRT (R)	R = INF or R = IND or R .LT. 0	+IND	39
SSWTCH (I1,I2)	I1 .GT. 6 or I1 .LE. 0	I2 = 2	40
TAN (R)	R = INF or R = IND or abs (R) .GT. 8.4E14	+IND	41
TANH (R)	R = INF or R = IND	+IND	42
OVERLAY	Fatal error reported by LOADER	Fatal	50
SEGMENT	Fatal error reported by LOADER Non-fatal error reported by LOADER	Fatal Proceed	51 52

Routine	Condition**	Standard Recovery	Error Number
BACKSP	Unassigned medium*	FATAL	53
BUFFEI	Unassigned medium*	FATAL	54
	Attempt to read past EOF on Buffer In.	FATAL	55
	Last operation was a write, no data available to read.	FATAL	56
	Starting address greater than terminal address.	FATAL	57
BUFFEO	Unassigned medium*	FATAL	58
	Starting address greater than terminal address.	FATAL	59
ENDFIL	Unassigned medium*	FATAL	60
IFENDF	Unassigned medium*	FATAL	61
INPUTB	Unassigned medium*	FATAL	62
	Attempt to read past EOF - coded input.	FATAL	65
INPUTS	Attempt to transfer more than 150 characters/rec. on DECODE processing.	FATAL	66
IOCHEK	Unassigned medium* for IF UNIT statement.	FATAL	67
KODER (Coded output)	Illegal Letter used as format specification.	FATAL	68
	Format specification with more than 2 levels of parentheses (3 levels under ASA).	FATAL	69

* The execution time diagnostic "Unassigned medium" is a result of a variable file name being undefined. The diagnostic printed out is actually "Unassigned medium," file xxxxxxx (where xxxxxxx is the name of the undefined file)

**All Input/Output errors at execution time are fatal errors. Therefore the standard error recovery for all of the above cases is (after standard error tracing is provided) to abort the job.

Routine	Condition**	Standard Recovery	Error Number
	Coded write past end of record.	FATAL	70
	Field width specified as zero.	FATAL	71
	Field width specified is less than or equal to the specified decimal width.	FATAL	72
	Attempt to output data under Hollerith format.	FATAL	73
KRAKER (Coded input)	Illegal letter used as format specification.	FATAL	74
	Format specification with more than 2 levels of parentheses.	FATAL	75
	Field width specified as zero.	FATAL	76
	Coded read past end of record.	FATAL	77
	Illegal data in the external field.	FATAL	78
	Data converted is out of range.	FATAL	79
	Attempt to input data under Hollerith format.	FATAL	80
LENGTH	Unassigned medium*	FATAL	81
OUTPTB	Unassigned medium*	FATAL	82
OUTPTC	Unassigned medium*	FATAL	83
	Line limit as specified on RUN card exceeded.	FATAL	84
OUTPTS	Attempt to transfer more than 150 characters/record on ENCODE processing.	FATAL	85
REWINM	Unassigned medium*	FATAL	86
KODER (Coded output)	Attempt to output a single array under "D" format specification.	FATAL	87

* The execution time diagnostic "Unassigned medium" is a result of a variable file name being undefined. The diagnostic printed out is actually "Unassigned medium," file xxxxxxx (where xxxxxxx is the name of the undefined file).

**All Input/Output errors at execution time are fatal errors. Therefore the standard error recovery for all of the above cases is (after standard error tracing is provided) to abort the job.

LIMITATIONS AND KNOWN DEFICIENCIES

1. The following description of BUFFER IN/BUFFER OUT is intended to clarify their use.

   When a BUFFER IN is performed on any medium, besides BCD 1/2-inch tape, one and only one logical record is read each time BUFFEI is called. If the block length specified by the call is longer than the logical record, the excess block locations will not be changed by the read. If the logical record is longer than one block, the excess words in the logical record are passed over. They will be counted but not transmitted to the program area. The number of central memory words in the logical record may be obtained by referencing LENGTH.

   Since there is no logical record concept per se on BCD 1/2-inch tape, the above must be modified slightly. In this case as many 136-character physical records will be read as is necessary to fill the block. If not all of the last physical record read is needed, the physical record will be passed over and counted, but the excess words will not be transmitted to the program area.

   When a BUFFER OUT is performed on any medium, besides BCD 1/2-inch tape, one logical record is written each time the routine is called. The record consists of a number of standard physical records (the size depending upon the medium) and a short record, or just a short record if the block length is less than the physical record size.

   For BCD output on 1/2-inch tape the record consists of 136-character physical records. If the block length is less than 136 characters the physical record is blank-filled to 136 characters. If the block length requires several physical records the last record is blank-filled to 136 characters if necessary.

   There are two restrictions on the use of BUFFER IN/BUFFER OUT:

   a. BUFFER IN does not read a mixed mode (both binary and BCD) file on 1/2-inch tape, although BUFFER OUT will write it. Parity errors occur while reading tape.

   b. When buffering out more than 136 characters on BCD 1/2-inch tape every fourteenth word loses its last 4 characters. Since the tape driver is designed to write only 136-character records, it picks up 14 words and writes the first 136 characters; the second physical record begins with the fifteenth word. When buffering in more than 136 characters the same process occurs. Every fourteenth word is zero-filled for its last 24 bits. The number of central memory words obtained by referencing LENGTH counts every physical record as 14 words.

2. The 64/6600 FORTRAN Version 2.0 compiler has been corrected through PSR number 213 with the following exceptions: 13, 208.

3. The 64/6600 FORTRAN Version 2.0 object time routines have been corrected through PSR 213 with the following exceptions: 162, 192.

4. The following DO loop compiles incorrectly if A and Z are TYPE COMPLEX.

```
DO   1   I = 1,3
DO   1   J = 1,3
A (I,J) = Z (I,J) ** K
```

5. A DATA declaration such as the following, which attempts to store excess elements into an array, receives a non-fatal **DR** diagnostic, as per specifications. However, an arithmetic error may occur during the compilation of the program.

```
DIMENSION A (3)
DATA A/1., 2., 3., 4./
```

6. If an ASCENT subroutine precedes the main FORTRAN program, the job will abort at execution time with a buffer argument error.

7. DBAIEX does not do the published error testing.

8. Non-standard error recovery for the ** power routines is of limited usefulness because the arguments of the routines are not available to the non-standard recovery, and the contents of A0 is destroyed in transferring to the non-standard recovery. Therefore, return should not be made to the place the power routine was called.

9. The interpretation of the fourth argument in calls to SEGMENT is backwards. If the argument is zero, unsatisfied externals are not satisfied from the library, if it is non-zero, they are satisfied from the library.

10. A mode 4 arithmetic error may occur in EXP for a very large negative argument. (This is also present in the 1.1 EXP routine.)

11. DATA statements of the CDC form:

    DATA (l = n,n),..., (m = n,n...n)

    are illegally flagged if a complex constant is the last element in a parenthetical group other than the last group. For example:

    DATA (C = (1.,2.)), (R = 3.), where C is complex

    does not compile whereas:

    DATA (A = 1.), (C = (1.,2.))

    does compile. (This also happened in Version 1.1.)

12. Inaccuracies in the results of calls to TAN may occur if the result is very large. (This is also present in the Version 1.1 TAN routine.)

13. The format conversion specification:

    k (nH...) or k (*...*)

    fails if the associated PRINT or WRITE statement contains no list or if the above specification appears after all list elements have been converted. For example:

```
          WRITE (10,500)
500       FORMAT (50(1H*))
```

    outputs only one asterisk whereas:

```
          WRITE (10,500) A
500       FORMAT (1X,3(1H*), E10.3,5(2HXX))
```

    terminates output with only one of the five "XX".

14. End-file marks are not always detected if both binary and BCD operations have been performed on the same unit. In the following example, the end-file at statement 500 is not detected.

```
          WRITE (10,1) A
          ENDFILE 10
          WRITE (10) I
          ENDFILE 10
          REWIND 10
          READ (10,1) A
          READ (10)
500       IF (EOF, 10) 2,3
```

15. The printer carriage control character + suppresses spacing after, rather than before, printing.

16. BUFFEI does not give a diagnostic when an attempt is made to read past an EOF. Every BUFFER I/O operation must be followed by an IF (UNIT, i) statement to check this.

17. The use of an erroneous file name in the RUN card causes the compiler to revert to the system name which would otherwise have been assumed for that parameter.

LIMITATIONS AND KNOWN DEFICIENCIES

1. Erroneous completion dates are entered for the beginning event of the network if conflicting actual and scheduled dates are input. This can be corrected by the user by removing the scheduled date on the beginning event.

SCOPE Reference Manual, Pub. No. 60173800

Page 3-2 Under PUNCHB change the fifth line of the example to read:

```
COPYBR (DAYFILE,A)
```

Page 7-10 The following error messages replace those in Section 7.3.8:

GF1 A PARAMETER IS GREATER THAN 7 CHARACTERS

The first separator or parameter terminator appears after eight alphanumeric characters. GF1 can appear for any of the three parameters.

GF2 A NUMERIC EXTENDS BEYOND AN END OF FILE

P2 is numeric and is too large. The double end of file is reached before P2 is satisfied.

COPYN writes all the existing records, one end of file, and then rewinds the file.

GF3 AN ID(P1) IS REQUIRED ON ALL TEXT CARDS

A comma or separator is the first character, causing the first parameter to be a zero.

GF4 TEXT CARD CONTAINS AN ILLEGAL SEPARATOR

Only , . blank + - / * are acceptable in addition to the alphanumeric characters.

GF5 CONTROL CARD REWIND (INPUT) IS ILLEGAL

COPYN could not reposition INPUT. Therefore the card is rejected and the message printed. INPUT is left unchanged.

GF6 TOO MANY INPUT FILE NAMES ON COPYN

The current limit is ten files.

COPYN gives an error message, attempts to use the first 10 parameters, and begins execution of the program.

GF7 NO OUTPUT FILE ON THE COPYN CONTROL CARD

The second parameter on the COPYN control card is zero. COPYN sets a disk file, TEMP, as the output file and continues to process the control card.

GF8 FIELD IS NON NUMERIC ILLEGAL TEXT CARD

The SKIPR and SKIPF requests cause this error message to be given when I is not numeric.

GF9 NO INPUT FILE ON THE COPYN CONTROL CARD

Parameters three through ten on the COPYN card are zero. A disk file, TEMP, is set as the only file searched when P3 is zero (exception—an existing P3 will be searched first).

GF10 BINARY RECORD MISSING FROM INPUT

P3 is INPUT and the next record on INPUT is not the expected binary record.

GF11 ID NAME NOT IN INPUT FILES SEARCHED

The P1 parameter was not found in either P3 or any of the input files listed on the COPYN control card.

GF12 TOO MANY TEXT CARDS IN THE INPUT RECORD

BUFF is the size of the input buffer. If there are more TEXT cards than are allocated by BUFF, all the cards in the buffer are processed, then the error message is printed.

GF13 P2 IS NOT IN THE FILE OR IS UNDEFINED

Either P2 was not found in the file or it began with * or /. P2 was not *, ** or /.

GF14 A DOUBLE EOF WAS FOUND BEFORE A /

When P2 is a / and the end-of-file is encountered before a zero length record, G14 is printed and all records to the EOF are written on the output file.

GF15 A PARAMETER BEGINS BEYOND AN EOF-EOF

P1 is numeric and causes a skipping to the double end of file before P1 is satisfied. The tape is positioned before the double end of file.